REAL-TIME AND EMBEDDED SYSTEMS BUILDING BLOCKS FOR CYBER-PHYSICAL SYSTEMS

BY

QIXIN WANG

B. Eng., Tsinghua University, 1999
B. Econ., Tsinghua University, 1999
M. Eng., Tsinghua University, 2001

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2008

Urbana, Illinois

Doctoral Committee:

Professor Lui Sha, Chair
Professor P. R. Kumar
Assistant Professor Marco Caccamo
Assistant Professor Rong Zheng, University of Houston

Abstract

By converging the cyber world with the physical world, Cyber-Physical Systems (CPS) is expected to create a great impact on computer science and the society. This thesis proposes several real-time building blocks for CPS infrastructures.

As for real-time wireless LAN, I compare various wireless communication paradigms and show that by deploying largest processing gain possible (i.e. lowest bit rate), DSSS-CDMA cell phone paradigm (i.e. each control loop occupies a CDMA channel between the remote station and the base station) achieves much higher reliability/robustness than main stream alternatives, such as IEEE 802.11, IEEE 802.15.4, and convolutional coding. For example, if a control loop sends one packet per second with a packet size of 152 bits, a DSSS-CDMA communication link using largest processing gain possible takes about 20dB more power to jam compared to an IEEE 802.11b link using packet retransmission. If the control loop sends ten packets per second, a DSSS-CDMA communication link is about 10dB more robust than an IEEE 802.11b link.

As for real-time wired network, I propose a real-time crossbar switch design, which is not only compatible to, but also simpler than the main stream iSLIP switch design. Specifically, since real-time flows are predetermined and run consistently for long time, an iSLIP switch can be easily revised into a real-time switch: we only need to grant according to a deterministic schedule; and eliminate the request and accept steps. The schedule can be represented as an N by M matrix S, where N is the number of input(output) ports of the crossbar switch, and M cell-time is the period of the schedule. Let S(i,k) denote the element at the ith row and kth column of S. Then at the kth cell-time of each period, the ith input port forwards a cell to the S(i,k)th output port. Basically, this means the switch serves each real-time flow with clock-driven time-slicing. A flow of period P cell-time and message size of C cells are over-provisioned with a server that forwards ceil(C/floor(P/M)) cells every M cell-time. By mapping each real-time flow to a server, we derive an N by N demand matrix D. The element at the ith row and jth column of D, denoted as D(i,j), means every M cell-time, input port i has to forward D(i,j) cells to output j. We prove that as long as each input (output) is required to forward (receive) no more than M cells per M cell-time period, demand matrix D can always be scheduled within polynomial time (O(power(N,4))).

I also designed a hard real-time, fast, and lightweight acoustic event localization protocol, the Lightning Protocol, for wireless sensor networks. Basically, wireless sensors are deployed in a square grid pattern. Every sensor is colored i (i = 1, 2, 3, 4), so that for any point on the plane, the enclosing four sensors have distinct colors. A sensor is either in RF listening or broadcasting mode (never both). Whenever a free (i.e. its state is neither winner nor loser) sensor hears the acoustic event, it broadcasts a noise on the wireless carrier for i time units. After that, if it does not hear any other noise on the wireless carrier, it wins the election; otherwise it loses the election. Whenever a free sensor hears a noise on the wireless medium, it loses the election. I prove this protocol elects the closest sensor with only O(1) RF broadcast within O(1) time. Energy Efficient Lightning Protocol is also designed, which only turns on RF module during localization period. Experiments using U. C. Berkeley Mica Motes show the feasibility of the protocol in lab environments.

To my parents: Jianren Wang and Yiying Wang

In memory of Prof. Jennifer C. Hou

Acknowledgments

The completion of this disseration and the underlying body of research during my Ph.D. study would not have been possible without the advice and cooperation of a number of people. The most important contribution has undoubtedly been due to my adviser, Prof. Lui Sha. Without his vision, it is highly unlikely that I would have chosen the meaningful topics presented in this dissertation. In the past seven years, Prof. Sha gave me critical and substantial advice, support, and freedom to explore the grandeur of academic world. Most importantly, he taught me fine taste and philosophy to do top-class research, which, I deeply believe, is one of the most precious gifts I can get in my entire life.

I am also grateful to Prof. Marco Caccamo, Prof. Jennifer Hou, Prof. P. R. Kumar, and Prof. Rong Zheng for serving on my thesis committee, for their kind comments, and for their help throughout the past seven years. I am especially grateful to Prof. Jennifer Hou, who made significant amount of effort to help advance my career, even when her health condition was deteriorating. I am also thankful for other professors in UIUC and other institutes for their enlightening advice and crucial help, including Prof. Tarek Abdelzaher, Prof. Sathish Gopalakrishnan, Prof. Carl A. Gunter, Prof. Elsa Gunter, Prof. Christophorus Hadjicostas, Prof. Bruce Hajek, Prof. Karrie Karahalios, Prof. Chang-Gun Lee, Prof. Xue Liu, Prof. Venugopal V. Veeravalli etc.

A number of my fellow students and colleagues have been instrumental in making fruitful the last seven years of my life, academically and otherwise. Special thanks to Bedoor K. Alshebli, Chin-Fei Cheah, Wei-Peng Chen, Weiqun Chen, Tanya L. Crenshaw, Eric Gilbert, Wenbo He, Jin Heo, Kihwal Lee, Min-Young Nam, Cham Oh, Wook Shin, Mu Sun, Ajay Tirumala, Yang Yu, and Zheng Zeng for co-authoring and reviewing my papers. And last but not least, thanks to the members of Real-Time Systems Lab, both real and honorary. Without the constant companionship provided by this diverse and interesting group of people, my Ph.D. study would have been far less memorable.

Table of Contents

Chapter 1

INTRODUCTION

By converging individual computers, the Internet has fundamentally changed our ways of life. To take one step further, *Cyber-Physical Systems* (CPS) [1–4] aims to converge the cyber world, where information is exchanged and exploited, and the physical world, where we live. This will create a grand impact on our society, enabling a vision that was unimaginable before. The following lists some example CPS applications.

1. Telepresense

As Adam Smith pointed out, specialization keeps advancing throughout the history, and along with it is the forever increasing need for collaboration. This rule results in the myriad of nowadays mega-cities and is responsible for a large portion of today's travels: people have to live close to each other and commute/travel to the same place to collaborate. Yet this solution is not sustainable: nowadays mega-cities are already struggling with serious pollution, exhaustion of space, traffic jam, and waste of non-renewable fossil energy. According to Time [5], 88% of all trips in the U.S. are by car; and according to the Department of Energy, transportation constitutes 28.4% of the United State's energy cost in 2006, reaching a highest share recorded since 1970 [6]. All of the above add up to an serious social economic challenge, especially with the recent development on global warming and energy shortage.

A hope to address this challenge lies in telepresence, which lets people work at home and collaborate through the next generation CPS network. For example, a construction worker can work at home by remotely controling a robot at the construction site, while the robot feeds back visual, acoustic, and haptic senses to the worker. In this way, people do not need to commute/travel to the same geographical location to collaborate, and the demand for mega-city can be relieved.

Yet this is way from all what telepresence can contribute. Two special cases of telepresence, telemedicine and tele-robotic underground mining, have other important social impacts respectively. Telemedicine can extend high quality medical care to much larger population, especially people living in remote areas and developing countries. Tele-robotic underground mining can help eliminate the over 5000 annual underground

mining death toll by replacing all underground personnel with robots operated from up-ground.

2. Next Generation Industrial Control

There are three trends in the next generation industrial control: more intensive use of intelligent embedded computer systems, increased interaction and complexity, and the adoption of wireless communications for mobility, flexibility, and other benefits.

The increased use of embedded systems results in increased interaction and complexity, which in turn call for a more comprehensive communication infrastructure for industrial control. This infrastructure shall consists of a wireless last hop and a wired backbone. A wireless last hop can increase mobility, enhance flexibility, and promote safety (e.g., tele-robotics in hazardous areas). A wired backbone provides better reliability and robustness, supports larger throughput, and reuses legacy infrastructures in the factories.

3. Next Generation Vehicle

Similar to next generation industrial control, next generation vehicles, such as airplanes, ships, and automobiles, also face the trend of increased use of embedded computing systems and communication complexity. One trend is that the bus architecture (e.g. PCI bus) will evolve to switch architecture, such as InfiniBand [7], to support parallelism between CPU and its peripherals.

4. Medical Device Plug-and-Play

Nowadays hospitals use tens of thousands, or even hundreds of thousands of electronic devices manufactured by a myriad of vendors. Currently, most of these devices are designed to run exclusively. However, there is great demand to integrate all devices into an open architecture, to allow more comprehesive applications. The *Medical Device Plug-and-Play* (MDPnP) [8] effort is one of the results of this demand.

One specific problem on MDPnP is operation room automation. Nowadays operation rooms are typically equipped with tens of medical devices, wired with cables. This presents a hazardous working environment (see Fig. 1.1 (a), where medical personnel may trip over the cables and cause accidents. In fact, healthcare professionals are explicitly requesting to "use wireless technologies to eliminate the 'malignant spaghetti' of cable clutter that interferes with patient care, creates hazards for the clinical staff and delays positioning and transport" [9] (see Fig. 1.1(b) for a wireless operation room).

5. Assisted Living

The aging of baby boomers is creating social and economic challenges. In the United States alone, the number of people over age 65 is expected to hit 70 million by 2030, doubling from 35 million in 2000. As the population ages there will be an increasing demand on health care resources. Approximately one-third of the health care expenditures are directed to the population over age 65. The fastest growing segment of the older adult population is those over age 85. This population will also double by 2030 [10]. Almost 25% of

(a) A wired operation room

(b) A wireless operation room

Figure 1.1: Wired and wireless operation room (pictures quoted from presentation by Dr. Julian Goldman at NSF High Confidence Medical Device Software and Systems Workshop

this population have one or more deficits in their activities necessary for successful daily living (i.e., dressing and bathing) and resides in a skilled or assisted living facility. Because of the increasing demands of our aging population on the health care system, expenditures in the United States for health-care will grow to 15.9% of the GDP ($2.6 trillion) by 2010 (Digital 4Sight's Healthcare Industry Study) [11] . The ability of the current system to shift the burden of care to the family members will become increasingly limited, as a result of the decrease in the birth rate and the increasing number of adults surviving to old age without living children. These same demographics contribute to the declining proportion of our society in the work force that must support the increasing health care costs of those who have retired [12]. Clearly the current system of health care delivery is not sustainable. Innovative strategies will be needed to avoid the impending crisis. One way out is to integrate various embedded devices, such as wireless sensors, RFID, and bluetooth

vital sign monitoring devices to create a CPS assisted living environment, which helps elderly people to live independenlty.

The list of CPS applications can go on. But to support these applications, we must first design the corresponding building blocks to lay the infrastructure. One large category of these building blocks are the real-time and embedded systems building blocks.

For example, we need a combined wireless and wired real-time network infrastructure to support telepresence, next generation industrial control, and MDPnP. We want real-time wireless LANs at the last hop for better mobility and flexibility; and want real-time wired WAN as the backbone for reliability, throughput, and legacy reuse considerations. For real-time wireless LAN, the top concern resides in communication reliability/robustness. On the one hand, wireless communication by nature is less reliable/robust than wired communication due to large-scale path loss, multi-path, and *Radio Frequency* (RF) jamming. Many CPS environments, such as hospitals, underground mines, and factories deteriorate the situation: heavy obstructions increase large-scale path loss and multi-path; *Electro-Magnetic Interference* (EMI) from electric motors, welding, and power cords persistently jam the wireless channel; and non-cooperative RF devices can be accidentally or macliciously turned on to create additional jamming. On the other hand, real-time requirements mendate wireless connections to work continuously even under adverse channel conditions; backoff based MAC is not allowed. For wired WAN backbone, the main challenge lies in the switch (a.k.a. router) design for efficient real-time packet forwarding. The success of such design largely depends on its backward compatibility with nowadays main stream switch designs, which, however, are not for real-time.

In contrast to the above combined wireless/wired real-time networks, *Wireless Sensor Networks* (WSN) are another form of real-time/embedded systems. WSNs need many basic system services, one of which is real-time acoustic event localization. Real-time acoustic event localization plays a key role for many CPS applications, such as assisted living, environment monitoring, and security. Yet how to design a hard real-time, fast, and lightweight acoustic event localization mechanism for low-end wireless sensors is another challenge.

We will address these aforementioned challenges in the following chapters. In Chapter 2, we compare various wireless communication paradigms and argue that by deploying largest processing gain (lowest data rate), DSSS-CDMA cell phone paradigm achieves much higher reliability/robustness than main stream alternatives, such as IEEE 802.11, IEEE 802.15.4, and convolutional coding. In Chapter 3, we propose a wired WAN real-time switch design, which is not only compatible with, but also simpler than the main

stream *i*SLIP [13,14] switch design. In Chapter 4, we propose a hard real-time, fast, and lightweight acoustic event localization protocol that elects the closest sensor to an acoustic event with only $O(1)$ RF broadcasts. We conclude the dissertation with Chapter 5, which also discusses our future research directions.

Parts of this thesis have been previously published in the following papers:

1. Copyright© 2007 IEEE. IEEE Transactions on Mobile Computing, volume 6, issue 6, pp. 706-719, "Building Robust Wireless LAN for Industrial Control with the DSSS-CDMA Cell Phone Network Paradigm," Qixin Wang, Xue Liu, Weiqun Chen, Lui Sha, and Marco Caccamo.

2. Copyright© 2008 IEEE. Proceedings of the 14th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2008), "A Switch Design for Real-Time Industrial Networks," Qixin Wang, Sathish Gopalakrishnan, Xue Liu, and Lui Sha.

3. Copyright© 2008 IEEE. IEEE Transactions on Mobile Computing, volume 7, issue 5, pp. 570-584, "Lightning: A Hard Real-Time, Fast, and Lightweight Low-End Wireless Sensor Election Protocol for Acoustic Event Localization," Qixin Wang, Rong Zheng, Ajay Tirumala, Xue Liu, and Lui Sha.

4. Copyright© 2006 IEEE. Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (ICSMC'06), v5, pp. 4268-4275, "I-Living: An Open System Architecture for Assisted Living," Qixin Wang, Wook Shin, Xue Liu, Zheng Zeng, Cham Oh, Bedoor K. Alshebli, Marco Caccamo, Carl A. Gunter, Elsa Gunter, Jennifer Hou, Karrie Karahalios, and Lui Sha.

Chapter 2

ROBUST/RELIABLE REAL-TIME WIRELESS LAN

Real-time wireless communication infrastructure is important for many CPS applications, such as telepresence, next generation industrial control, and operation room automation. Particularly, people are interested in designing and deploying *Real-Time Wireless LANs* (RT-WLAN) for CPS applications [15–24], so as to increase mobility, reduce deployment cost, enhance flexibility, and promote safety (e.g., remote control of robots in hazardous areas).

Nonetheless, a major concern of RT-WLAN is its robustness/reliability: wireless communication must be maintained under adverse channel conditions. Wireless channel conditions are inherently more vulnerable than those of wireline communications for the existence of such problems as multiple-access contention, *Radio Frequency* (RF) interference, large-scale path loss, and fading (a.k.a. multipath) [25]. Industrial environments make these problems deteriorate because of heavy obstructions [25] and possible *Electro-Magnetic Interferences* (EMIs) [26, 27]. An example is that EMI from electric welding or electric motor can last for hours, or even days. Nevertheless, RT-WLANs require much higher robustness than conventional WLANs for office or home use. Most office or home wireless communications allow a few seconds or even minutes of adverse channel conditions. They just need to backoff till the channel condition recovers, and then retransmit. Industrial control, however, often forbids such backoff behavior. Because most industrial control loops are real-time, the backoff behavior will cause deadline misses, which further trigger performance losses, halts/resets of manufacturing pipelines, or defects in products. For example, 200msec of backoff may incite an inverted pendulum [28] fall. Therefore, for most industrial controls, communications must be maintained even under adverse channel conditions instead of backing off.

RF interference, large-scale path loss and fading cause adverse channel conditions by reducing *Signal-to-Noise Ratio* (SNR) of the wireless communications. When the SNR is lower than a certain threshold, the *Bit Error Rate* (BER) of the wireless communication rises over the acceptable limit, thus disrupting the wireless connection. Therefore, the key to maintaining wireless communication under adverse channel conditions is to provide as high SNR as possible. To achieve this, a promising solution lies in the state-of-the-art *Direct Sequence Spread Spectrum* (DSSS) technology, which allows tradeoffs between data throughput versus SNR.

Specifically, a lower data throughput corresponds to a higher SNR and vice versa. Fortunately, industrial control loop traffics in RT-WLANs are often low-data-throughput stable traffics [29]. For example, most industrial mechanical systems carry out fine-grained high-rate controls locally using step motors [30–32], so that only low-data-throughput coarse-grained control traffics are transmitted between distributed nodes. Typically, the sampling/actuating rates between distributed nodes are around $1 \sim 10$Hz, and the packet sizes are around $100 \sim 200$ bits.

Based on above observations, we propose using DSSS technology to fully exploit the low-data-throughput feature of control loop traffics, to build robust RT-WLANs. Through fine-grained physical layer simulations and Monte Carlo comparisons, we show that when the low-data-throughput feature is fully exploited, DSSS RT-WLANs achieve much higher robustness than IEEE 802.11/802.15.4 WLANs (for consistency, we refer to IEEE 802.15.4 as a WLAN scheme in this paper) [33–35] do, so that wireless industrial control becomes practical (see Section 2.3). Specifically, a DSSS RT-WLAN achieves $10 \sim 20$dB and $20 \sim 30$dB improvements on robustness compared to an IEEE 802.11b and an IEEE 802.11a WLAN respectively; similar improvements are also achieved against IEEE 802.15.4 WLANs. These are significant improvements according to communication engineering criteria.

DSSS is a physical layer scheme, which only concerns point-to-point communications. At the *Multiple Access Control* (MAC) layer, we need a proper RT-WLAN paradigm, which can either be the fully distributed ad hoc paradigm of IEEE 802.11/802.15.4 WLANs, or the centralized *Code Division Multiple Access* (CDMA) paradigm of cellphone networks. We prefer the CDMA cellphone network paradigm. Under such paradigm, every RT-WLAN is a cell, with one *base station* and several *remote stations*; wireless communications only take place between a base station and a remote station of the same cell; inter-cell communications only exist between base stations via wireline backbones. The reasons why CDMA cellphone network paradigm is preferred run as follows: i) Industrial control loop traffics are usually real-time. The base-station-centered CDMA cellphone network paradigm makes it easy to implement centralized real-time scheduling. In practice, centralized real-time scheduling is often more desirable due to its robustness and simplicity. ii) Most industrial control loops incur low computation, therefore it is a common and economic practice to have one powerful centralized base station controlling all machines in a local area [29]. Many legacy systems are already built upon such base-station-centered communication paradigm. iii) Industrial control applications are typically deployed in well-built permanent facilities, where powerful wireline backbones for inter-base-station communications are available. Therefore, the benefits of wireless communications (mechanical freedom, mobility, flexibility) are only significant at the last hop. A CDMA cellphone network paradigm matches such need. iv) CDMA is also a more preferable technology due to its ease of scheduling,

overrun isolation and low communication overhead.

To sum up, this paper mainly demonstrates that by fully exploiting the low-data-throughput feature of industrial control loops, the DSSS-CDMA cellphone network paradigm presents a better approach to build robust RT-WLANs than the nowadays predominant IEEE 802.11/802.15.4 paradigms. This paper also studies some resource management issues on the proposed DSSS-CDMA RT-WLAN. As an example, we derive optimal resource configuration for maximal robustness. The resource management issues open a new problem space for interdisciplinary study, which involves real-time scheduling, communication, networking and control.

The rest of the chapter is organized as follows: Section 2.1 gives background on DSSS technology. Section 2.2 proposes the DSSS-CDMA RT-WLAN scheme, together with some analytical results on its resource optimization. Section 2.3 carries out fine-grained physical layer simulations to demonstrate DSSS-CDMA RT-WLAN robustness, and more extensive Monte Carlo simulations to compare the robustness with IEEE 802.11/802.15.4 WLANs'. Section 2.3 also includes a discussion on the feasibility of error correction coding besides DSSS. Section 2.4 discusses related works. Section 2.5 concludes the chapter.

2.1 Background

DSSS is a physical layer modulation/demodulation scheme for digital communication [36–38]. It modulates/demodulates the original data signal to/from a baseband signal which occupies a wider spectrum[1]. At the transmitter, a user data bit stream of bit rate r_b (every bit takes $T_b \stackrel{def}{=} 1/r_b(\text{sec})$) is *scrambled* with a *Pseudo Noise* (PN) sequence of *chip rate* r_c (every chip takes $T_c \stackrel{def}{=} 1/r_c(\text{sec})$), producing a chip stream of rate r_c. r_c is a positive integer multiple of r_b, the ratio $g \stackrel{def}{=} r_c/r_b$ is called *processing gain*. At the receiver, if the chip stream is *descrambled* with the same PN sequence, the original data bit stream recovers. If a different PN sequence is applied or the scramble/descramble PN sequences are not synchronized, the original data bit stream does not recover and a noise-like random chip stream is generated instead. To summarize, each PN sequence creates a DSSS data channel. Note although DSSS requires synchronization between each transmitter and its receiver, different transmitter-receiver pairs need not be synchronized. Appendix I of [40] gives a more detailed tutorial on DSSS.

DSSS is a physical layer scheme. At the MAC layer, there are two alternatives: *Code Division Multiple Access* (CDMA), or *Time Division Multiple Access* (TDMA). For simplicity, we also categorize the widely

[1]We refer to DSSS as a *baseband* modulation/demodulation scheme. In contrast, the modulation/demodulation scheme that shifts baseband signal to/from RF band is referred to as RF modulation/demodulation. Typical RF modulation/demodulation schemes for DSSS can be *Quadrature Phase Shift Keying* (QPSK) or *Binary Phase Shift Keying* (BPSK), both can achieve the same robustness (in sense of BER) with the same SNR per bit [36, 39].

used *Carrier Sensing Multiple Access* (CSMA) as a kind of TDMA. If DSSS-CDMA is deployed, different data bit streams scrambled with different PN sequences are transmitted in parallel through the same RF band. At each receiver, by applying different PN sequences, the intended data bit stream is filtered out. If DSSS-TDMA is deployed, different data bit streams are scrambled and transmitted in non-overlapping time slots. Though both alternatives work, DSSS-CDMA fits RT-WLANs better because: i) ease of real-time scheduling; ii) inherent isolation between connections; iii) less communication overhead, especially under adverse channel conditions. i) and ii) are straightforward and interrelated: Under DSSS-CDMA, a real-time connection exclusively occupies a CDMA channel by using a unique DSSS PN sequence. Different CDMA channels can coexist in parallel. Therefore it is not necessary to schedule different real-time connections, and the overrun of one real-time connection does not affect any other real-time connections. In contrast, under DSSS-TDMA, the DSSS PN sequence is shared among all real-time connections, and different time slots must be scheduled to serve different connections. If a real-time connection overruns its time slot, subsequent real-time connections are affected. In terms of iii), a simplified explanation is as follows: DSSS requires time synchronization between the transmitter and the receiver. Under CDMA, packets of a same connection are sent continuously as one bit stream (i.e., *session*). Synchronization time cost only happens during session setup. During the session, synchronization is maintained in parallel of data transmission. Under TDMA, however, every packet incurs synchronization time cost. Under adverse channel conditions, this cost may be big, causing much more overhead in TDMA than CDMA. Appendix II of [40] further elaborates this.

Quantitatively, many important features of DSSS are captured by its *Bit Error Rate* (BER) upper bound shown in inequality (2.1)[2] [36, 41]:

$$\mathcal{P}_{ber} \leq \exp\left(-\frac{gP_u}{J + \sum_{i=1, i\neq u}^{\Xi} P_i + \sum_{h=1}^{H} A_h + P_u}\right), \tag{2.1}$$

where $\mathcal{P}_{ber}$ is the BER; g is processing gain; J is the received power of *External RF Interference* (EI), which specifically refers to EMI, thermal noise and the RF interference from RF devices turned on accidentally or maliciously; P_i $(i = 1 \ldots \Xi)$ is the received power of CDMA channel i; Ξ is the total number of CDMA channels coexisting in parallel; u is the intended channel, with a received power of P_u. Each transmitting node may send out several CDMA channels in parallel, each carries a data stream. To facilitate the reception, the node may transmit an additional chip stream called *pilot tone* [36], which is synchronized with the node's outgoing data streams. In inequality (2.1) the pilot tone of transmitting node h $(h = 1, \ldots, H)$ is of power A_h. $\sum_{i=1, i\neq u}^{\Xi} P_i + \sum_{h=1}^{H} A_h$ is the upper bound of total *Multiple Access Interference* (MAI), that is, the

[2]Inequality (2.1) assumes QPSK RF modulation and per connection pilot tone. Different implementation alternatives may affect details of the inequality, though there will be no fundamental differences.

interference caused by other CDMA channels and pilot tones received in parallel with the intended channel. Note P_u also appears in the denominator, adding up to the total interference power. This is to provide a pessimistic estimation on *Inter Symbol Interference* (ISI), which usually results from multipath fading. To simplify, we can merge $\sum_{i=1,i\neq u}^{\Xi} P_i$ and P_u to be denoted as $\sum_i P_i$. The component $gP_u/(J+\sum_i P_i+\sum_h A_h)$ shows the effective SNR for the intended channel, $J+\sum_i P_i+\sum_h A_h$ representing the upper bound of noise power and gP_u representing effective signal power. *Inequality (2.1) implies the bigger the effective SNR, the smaller the probability of bit error $\mathcal{P}_{ber}$.*

A similar notion to BER is *Packet Error Rate* (PER). Without error correction coding, PER $\mathcal{P}_{per}$ is:

$$\mathcal{P}_{per} = 1-(1-\mathcal{P}_{ber})^{L^{pkt}} \tag{2.2}$$

$$\text{Or equivalently:} \quad \mathcal{P}_{ber} = 1-(1-\mathcal{P}_{per})^{1/L^{pkt}}, \tag{2.3}$$

where L^{pkt} is the bit length of the packet. When error-correction coding is deployed, equation (2.2) and (2.3) will have a more complicated form, but still, $\mathcal{P}_{per}$ and $\mathcal{P}_{ber}$ maintain one-to-one mapping and $\mathcal{P}_{per}$ decreases as $\mathcal{P}_{ber}$ decreases. When $\mathcal{P}_{per}$ is below a maximal acceptable threshold Θ_{per}, or equivalently, when $\mathcal{P}_{ber}$ is below a maximal acceptable threshold Θ_{ber}, the wireless communication is acceptable for industrial control. Remember inequality (2.1) implies the bigger the effective SNR, the smaller the BER. Therefore, maintaining an RT-WLAN wireless communication channel (i.e. to maintain $\mathcal{P}_{per} \leq \Theta_{per}$, or say, $\mathcal{P}_{ber} \leq \Theta_{ber}$) means maintaining the effective SNR of the intended channel beyond a threshold Θ_{snr}:

$$\frac{gP_u}{J+\sum_i^{\Xi} P_i+\sum_h^{H} A_h} \tag{2.4}$$

$$\geq \Theta_{snr} \tag{2.5}$$

$$= -\ln\Theta_{ber} \qquad \text{(because of (2.1))} \tag{2.6}$$

$$= -\ln\left(1-(1-\Theta_{per})^{1/L^{pkt}}\right) \quad \text{(because of (2.3)).} \tag{2.7}$$

Expression (2.4) is the effective SNR of the intended channel, which can be raised by increasing the processing gain g. Since $g \stackrel{def}{=} r_c/r_b$ and chip rate r_c is usually fixed due to multipath effect and hardware cost constraints [36, 42], raising processing gain g means slowing down user data bit rate r_b. *DSSS hereby provides a mechanism to leverage between SNR and data bit rate.*

2.2 DSSS-CDMA RT-WLAN Architecture

2.2.1 The Overall Architecture

Based on previous discussions, we propose building RT-WLAN with DSSS-CDMA cellphone network paradigm: Every RT-WLAN is a cell. Each cell has one *base station* and several *remote stations*. Base stations of different cells are connected via a wireline backbone. All inter-cell communications only go through this wireline backbone. Within a single cell, the base station communicates with its remote stations through wireless. There are no direct wireless communications between remote stations. In this paper, we focus on the single cell, in other words, the single RT-WLAN scenario. Fig. 2.1 illustrates the architecture of a single RT-WLAN.

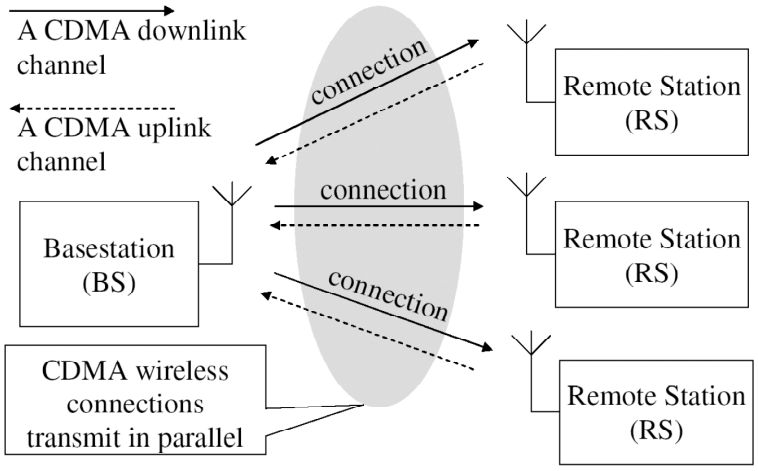


Figure 2.1: DSSS-CDMA RT-WLAN architecture

In an RT-WLAN, the RF band available is evenly partitioned into two halves: one for downlink (from base station to remote stations) and the other for uplink (from remote stations to base station). A wireless connection consists of one CDMA channel in each direction (downlink and uplink). Unless explicitly noted, "connection n", a.k.a. "control loop n", refers to both the downlink and uplink of the connection; "a CDMA channel of connection n" refers to both downlink and uplink CDMA channels of the connection. Without loss of generality, we assume sampling packets are sent in uplink, and actuating packets are sent in downlink. In each control loop, sampling/actuating packets are sent continuously so that their bits form a continuous uplink/downlink bit stream respectively. This also implies that the sampling/actuating period is the same as the packet transmission period.

2.2.2 Resource Planning for Maximized Robustness

In this paper, we attempt to analyze the optimal resource planning for maximized robustness: *given signal attenuation of every wireless connection, how to tolerate maximal external RF interference; and given external RF interference, how to tolerate maximal signal attenuation.* Here, "tolerate" means the packet error rate is maintained below the maximal acceptable threshold.

Although the derivation is complicated, the conclusion is simple and intuitive: maximal robustness is achieved when each connection deploys maximal possible processing gain (i.e., minimal possible data rate). This conclusion is formally described by the following proposition:

Proposition 2.2.1 (Maximal Robustness Configuration) *To achieve maximal robustness, control loop n ($n = 1, 2, \ldots, N$) of a DSSS-CDMA RT-WLAN should pick maximal possible processing gain:*

$$g_n^* = \min\left\{\left\lfloor r_c/(L_n^{pkt} f_n^{min})\right\rfloor, g^{max}\right\}, \tag{2.8}$$

where r_c is the fixed chip rate, L_n^{pkt} is the packet bit length of control loop n, f_n^{min} is the minimal allowed sampling/actuating rate of control loop n, and g^{max} is the maximal processing gain allowed by hardware.

To derive the above proposition, we first assume the RT-WLAN consists of control loop (a.k.a connection) 1, 2, ..., and N. Control loop n ($n = 1, 2, \ldots, N$) corresponds to a minimal sampling/actuating rate f_n^{min}, a maximal acceptable packet error rate Θ_n^{per}, and a sampling/actuating packet bit length L_n^{pkt} (assume sampling/actuating packets are of the same length; if not, paddings are used to make them the same). The two end nodes of control loop n are the base station, denoted as node 0, and a distinct remote station, denoted as node n. Every node of the RT-WLAN deploys a DSSS chip rate of r_c (i.e. a chip duration of $T_c = 1/r_c$), and carries out conventional *Quadrature Phase Shift Keying* (QPSK) RF modulation/demodulation. The per node maximal transmission power is P^{max}. In the uplink, power balancing is carried out to deal with near-far problem [25]. To assist reception, each node also transmits a pilot tone [36], whose allocated transmission power is the same as any of the node's outgoing CDMA data channels'. The transmitted signal attenuates in the wireless medium due to large-scale path-loss and fading. Let α_n^{down} and α_n^{up} denote the downlink/uplink attenuation of connection n respectively. At the receiver, transmitted signals are received together with external RF interferences (i.e. thermal noise, EMI, and malicious/accidental same-band RF device broadcasts). Denote J_n as the external RF interference power received at Node n ($n = 0, 1, 2, \ldots, N$, Node 0 refers to the base station, Node $1 \ldots N$ refer to remote station $1 \ldots N$ respectively).

The above parameters conform to following relationships:

The configurable parameters are the control loops' processing gain g_n $(n = 1, 2, \ldots, N)$, with the following value range:

$$1 \leq g_n \leq g^{max}, \text{ and } g_n \text{ is an integer.} \tag{2.9}$$

Here g^{max} is the maximal processing gain allowed by wireless communication hardware (e.g. if g_n is specified by an unsigned byte in the hardware, then g_n can not exceed 256).

Meanwhile, given chip rate r_c, packet bit length L_n^{pkt}, and the chosen processing gain g_n, the packet rate f_n (packets per second) is:

$$f_n = \frac{r_c}{g_n L_n^{pkt}}. \tag{2.10}$$

Remember the packet rate is the same as the sampling/actuating rate, which must satisfy the minimal sampling/actuating rate requirement, therefore:

$$f_n \geq f_n^{min} \Leftrightarrow g_n \leq \frac{r_c}{L_n^{pkt} f_n^{min}}, \qquad n = 1 \sim N. \tag{2.11}$$

In addition to constraints (2.9) and (2.11), there are two more constraints representing the robustness requirements: one for downlink; the other for uplink. *The robustness requirement for connection n is that the packet error rate should not exceed a maximal acceptable threshold Θ_n^{per}, both in downlink and uplink.* According to formulae (2.5) $\sim$ (2.7), maximal acceptable packet error rate Θ_n^{per} maps to a minimal acceptable SNR Θ_n^{snr}. That is, for each connection n, the effective SNR must be maintained above the minimal acceptable threshold Θ_n^{snr}. Inequality (2.5) quantitatively expresses this notion. According to inequality (2.5), the robustness requirement for downlink of connection n is:

$$\frac{g_n P_{nn}^{r\_dnlk}}{J_n + \sum_{i=1}^{N} P_{ni}^{r\_dnlk} + A_n^{r\_dnlk}} \geq \Theta_n^{snr}, \quad n = 1 \sim N, \tag{2.12}$$

where $P_{ni}^{r\_dnlk}$ $(i = 1, 2, \ldots, N)$ is the received power of CDMA downlink channel i at remote station n; $A_n^{r\_dnlk}$ is the received pilot tone power at remote station n (for downlink, base station is the only node that transmits pilot tone). Since the base station equally allocates its transmission power to all N downlink channels and the pilot tone, and the total transmission power of the base station is P^{max} to achieve maximal robustness, therefore:

$$\begin{aligned} & P_{n1}^{r\_dnlk} = P_{n2}^{r\_dnlk} = \ldots = P_{nN}^{r\_dnlk} \\ = \quad & A_n^{r\_dnlk} = \alpha_n^{down} \frac{P^{max}}{N+1}. \end{aligned} \tag{2.13}$$

Substituting formula (2.13) for (2.12), the downlink robustness requirement is converted to:

$$\frac{\alpha_n^{down} g_n P^{max}}{(N+1)(J_n + \alpha_n^{down} P^{max})} \geq \Theta_n^{snr}, \quad n = 1 \sim N. \tag{2.14}$$

For uplink of connection n, again according to inequality (2.5), the robustness requirement is:

$$\frac{g_n P_n^{r\_uplk}}{J_0 + \sum_{i=1}^{N} P_i^{r\_uplk} + \sum_{i=1}^{N} A_i^{r\_uplk}} \geq \Theta_n^{snr}, \quad n = 1 \sim N, \tag{2.15}$$

where $P_i^{r\_uplk}$ $(i = 1 \sim N)$ is the power of CDMA uplink channel i received at the base station; $A_i^{r\_uplk}$ $(i = 1 \sim N)$ is the power of remote station i's pilot tone received at the base station. Because of power balancing, there is:

$$P_1^{r\_uplk} = P_2^{r\_uplk} = \ldots = P_N^{r\_uplk}. \tag{2.16}$$

Meanwhile, assume each remote station i $(i = 1, 2, \ldots, N)$ equally divides its transmission power $P_i^{t\_uplk}$ between its uplink channel i and pilot tone, then:

$$P_i^{r\_uplk} = A_i^{r\_uplk} = \alpha_i^{up} \frac{P_i^{t\_uplk}}{2}, \qquad i = 1, 2, \ldots, N. \tag{2.17}$$

Also, each remote station cannot exceed its maximal transmission power:

$$P_i^{t\_uplk} \leq P^{max}, \qquad i = 1, 2, \ldots, N. \tag{2.18}$$

The transmission power of each remote station $P_i^{t\_uplk}$ should be maximized to increase SNR (and therefore robustness), meanwhile maintaining the constraints depicted in formulae (2.16) and (2.18). Therefore, the remote station that suffers the most severe uplink power attenuation should transmit with power P^{max}, and all the other remote stations should adjust their transmission power according to power balancing rule (2.16). The above is formalized as follows:

$$\begin{aligned} & P_k^{t\_uplk} = P^{max}, \\ & \text{therefore} \;\; P_k^{r\_uplk} = A_k^{r\_uplk} = \alpha_k^{up} \frac{P^{max}}{2}, \\ & \text{where} \;\; k = \text{argmin}_{i \in \{1,2,\ldots,N\}} \{\alpha_i^{up}\}. \end{aligned} \tag{2.19}$$

$$
\begin{aligned}
& (2.16), (2.17) \\
\Rightarrow \quad & P_i^{r\_uplk} = A_i^{r\_uplk} = \alpha_i^{up} \frac{P_i^{t\_uplk}}{2} = P_k^{r\_uplk} \\
\Rightarrow \quad & P_i^{t\_uplk} = \frac{\alpha_k^{up} P^{max}}{\alpha_i^{up}} \qquad \text{(because of (2.19))} \\
\Rightarrow \quad & P_i^{r\_uplk} = A_i^{r\_uplk} = \alpha_i^{up} \frac{P_i^{t\_uplk}}{2} = \alpha_k^{up} \frac{P^{max}}{2}. \qquad (2.20)
\end{aligned}
$$

Denote

$$
\alpha^{up} \stackrel{def}{=} \alpha_k^{up} = \min\{\alpha_1^{up}, \alpha_2^{up}, \ldots, \alpha_N^{up}\}, \qquad (2.21)
$$

and substitute (2.19) $\sim$ (2.21) for (2.15), the uplink robustness requirement is converted to:

$$
\frac{\alpha^{up} g_n P^{max}}{2(J_0 + \alpha^{up} N P^{max})} \geq \Theta_n^{snr}, \qquad n = 1, 2, \ldots, N. \qquad (2.22)
$$

The downlink/uplink robustness requirements (formulae (2.14) and (2.22)) can be converted as follows:

$$
\begin{aligned}
(2.14) \Leftrightarrow J_n \quad \leq \quad & \left(\frac{\alpha_n^{down} g_n P^{max}}{(N+1)\Theta_n^{snr}} - \alpha_n^{down} P^{max} \right) \\
& \stackrel{def}{=} \bar{J}_n^{down}, \quad n = 1, 2, \ldots, N; \qquad (2.23) \\
(2.22) \Leftrightarrow J_0 \quad \leq \quad & \left(\frac{\alpha^{up} g_n P^{max}}{2\Theta_n^{snr}} - \alpha^{up} N P^{max} \right) \\
& \stackrel{def}{=} \bar{J}_n^{up}, \quad n = 1, 2, \ldots, N, \qquad (2.24)
\end{aligned}
$$

where $\bar{J}_n^{down}$ and $\bar{J}_n^{up}$ represent the maximal tolerable external RF interference for downlink and uplink of connection n respectively. That is, when J_n exceeds $\bar{J}_n^{down}$, connection n's downlink will have a packet error rate over acceptable limit Θ_n^{per}; when J_0 exceeds $\bar{J}_n^{up}$, connections n's uplink will have a packet error rate over acceptable limit Θ_n^{per}. Define

$$
J^{min} \stackrel{def}{=} \min\{\bar{J}_1^{down}, \ldots, \bar{J}_N^{down}, \bar{J}_1^{up}, \bar{J}_2^{up}, \ldots, \bar{J}_N^{up}\}, \qquad (2.25)
$$

then J^{min} *represents the minimal external RF interference power needed to disrupt at least one of the connections.*

When the power attenuations α_1^{down}, α_2^{down}, ..., α_N^{down}, α_1^{up}, α_2^{up}, ..., α_N^{up} are given, robustness maximization means the RT-WLAN tolerates (i.e., the robustness requirements are satisfied, or quantitatively,

both inequality (2.23) and (2.24) sustain) maximal external RF interference power (i.e., J^{min} is maximized). Since the only configurable parameters are g_n ($n = 1 \sim N$), which comply with constraints (2.9) and (2.11), formulae (2.23) $\sim$ (2.25) imply that the RT-WLAN tolerates maximal external RF interference power when $g_n = \min\left\{\left\lfloor r_c/(L_n^{pkt} f_n^{min})\right\rfloor, g^{max}\right\}$. When $J_0, J_1, J_2, \ldots, J_N$ are given, robustness maximization means the RT-WLAN tolerates maximal power attenuations (i.e., α_1^{down}, α_2^{down}, ..., α_N^{down}, α_1^{up}, α_2^{up}, ..., α_N^{up} are minimized). Similarly, this is also achieved when $g_n = \min\left\{\left\lfloor r_c/(L_n^{pkt} f_n^{min})\right\rfloor, g^{max}\right\}$.

Therefore, Proposition 2.2.1 holds.

2.3 Simulation and Comparisons

In this section, we first demonstrate the robustness of the proposed DSSS-CDMA RT-WLAN based on fine-grained physical layer simulations. Then we carry out more comprehensive comparisons between the DSSS-CDMA scheme and conventional IEEE 802.11/802.15.4 schemes. In the end, the alternative of using error correction coding is discussed.

2.3.1 Demo Using Fine-Grained Physical Layer Simulation

We carry out fine-grained physical layer simulation to demonstrate the effectiveness of the proposed DSSS-CDMA RT-WLAN scheme. The simulation environment is built on top of J-Sim kernel [43]. Fig. 2.2(a) depicts the simulated scenario. According to it, the RT-WLAN includes two connections: connection 1 and 2. Each connection controls an *Inverted Pendulum* (IP) [28]: IP 1 and IP 2. As a remote station, each IP periodically sends back its state (x, θ, and time stamp) to the base station. Based on the most up-to-date IP state, the base station calculates the next control (u) and sends it back to the IP. The sampling/actuating packet length are both 152 bits, and the minimal sampling/actuating rates are $f_1^{min} = f_2^{min} = 10$Hz.

Without loss of generality, the two IPs are the same. As shown in Fig. 2.2(b), x is the position of IP cart, θ is the angular deviation of IP from vertical position, and u is the control voltage applied to IP cart. The state transition equation and control equation are also depicted in the figure (in addition, when θ and u are of opposite signs, u is obviously out-of-date due to delay and is therefore ignored). The IP cart moves along the x axis to keep the IP standing vertically. Each IP fall-down (defined as $|\theta|$ exceeds $\frac{\pi}{6}$) incurs a high cost resetting procedure.

We carry out simulation under both DSSS-CDMA and IEEE 802.11b schemes. A wireless medium instance generated from a typical indoor-industrial-environment model [25, 44] depicted in Table 2.1 is applied to the simulation of both schemes. To be fair, for both schemes, the maximal transmission power

Table 2.1: Wireless Medium Model

Large-scale path loss model	Log-normal shadowing model with $\beta = 4 \sim 6$, $\sigma = 6.8$dB *
Small-scale fading model	Rayleigh
Multipath max excess delay†	90.909nsec
Additive White Gaussian Noise‡	Spectral density = -174dBm/Hz

$*$ β is the path loss exponent, σ is the log-normal standard deviation.
† To deal with multipath fading, both DSSS-CDMA and IEEE 802.11b use two-finger RAKE receivers [36, 42].
‡ Typically refers to thermal noise.

of all nodes (the base station and all remote stations) are 1watt, the maximal transmission power allowed by FCC for IEEE 802.11b. The only exception is for DSSS-CDMA uplinks, where transmission power must also comply with the power balancing requirement to produce the same power level at the base station. The power balancing requirement makes the comparison more pessimistic on the DSSS-CDMA side, because some nodes are not transmitting with maximal power. Again, to make fair comparisons, both DSSS-CDMA and IEEE 802.11b schemes occupy the same RF band of $2.426 \sim 2.448$GHz, a typical RF band of IEEE 802.11b. For DSSS-CDMA, the RF band is divided into two halves: $2.426 \sim 2.437$GHz for downlink and $2.437 \sim 2.448$GHz for uplink. For IEEE 802.11b, the signal occupies the whole RF band, but packets are time multiplexed into downlink packets and uplink packets. These RF bandwidth configurations imply a chip rate of $r_c^{cdma} = 5.5$Mcps for DSSS-CDMA and a chip rate of $r_c^{ieee80211b} = 11$Mcps for IEEE 802.11b.

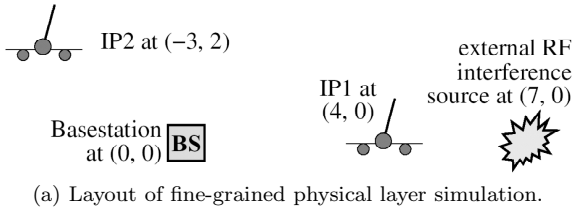
The above explains parameters relevant to both schemes. Further, the scheme-dependent details run as follows: For DSSS-CDMA scheme, the hardware-dependent processing gain upper bound g^{max} is 1024 (complies with cdmaOne [45]). According to Proposition 2.2.1, the processing gain that maximize robustness is therefore $g^{cdma} = \min\{\lfloor \frac{5.5\times10^6}{152\times10} \rfloor, 1024\} = 1024$. Without loss of generality, our DSSS-CDMA scheme deploys QPSK RF modulation/demodulation and per-node pilot tone. For each node, the pilot tone is allocated with the same transmission power as any outgoing CDMA data channel of the node. For IEEE 802.11b scheme, the most robust 1Mbps mode is deployed, corresponding to a processing gain of $g^{ieee80211b} = 11$ and *Differential BPSK* (DBPSK) RF modulation/demodulation. To be fair, the IEEE 802.11b WLAN works in pure PCF, the mode for real-time systems. Under PCF, the base station polls IP 1 and IP 2 in round robin without idling and backoff (backoff causes deadline miss). The control packet is sent to the IP as the poll packet, and the sample packet is sent back from IP as the acknowledgment packet.

To demonstrate the robustness of DSSS-CDMA scheme, an external RF interference source is placed near IP 1 (see Fig. 2.2(a)). This external RF interference occupies the same RF band that DSSS-CDMA and IEEE 802.11b are using. And the interference source transmits with a power of 1 watt, just the same

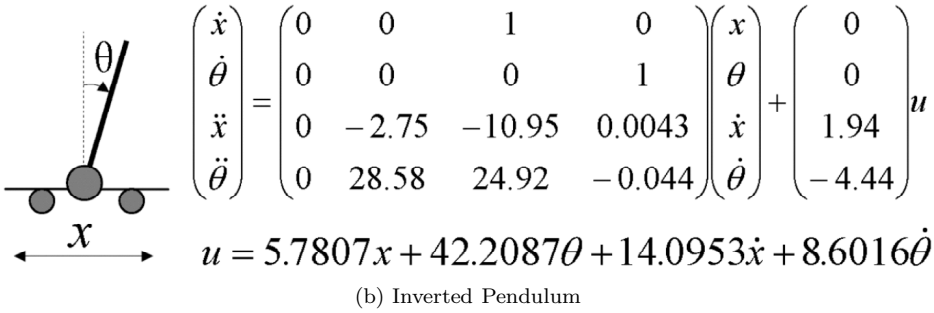
as the maximal transmission power of a normal RT-WLAN node.

The simulated scenario starts at time 0sec and ends at time 30sec. The external RF interference source is turned on at time 5sec and turned off at time 15sec.

Fig. 2.3 shows the traces of θ. The traces show that throughout the time, both IP 1 and IP 2 remain fairly stable under DSSS-CDMA, even when there is external RF interference ($5 \sim 15$sec). This means the wireless control loops survive adverse channel conditions. Under IEEE 802.11b, however, IP 1 keeps falling due to external RF interference (every time it falls, the IP resets to 0.5rad and stays there for 0.2sec to restart). Note under IEEE 802.11b, IP 2 can also survive external RF interference because it is much closer to the base station than to the external RF interference source.



(a) Layout of fine-grained physical layer simulation.



(b) Inverted Pendulum

Figure 2.2: Simulated Scenario

2.3.2 Comparisons to IEEE 802.11 a/b

IEEE 802.11 is the nowadays predominant WLAN scheme. It can be further categorized into IEEE 802.11b [46], a [47] and g [48]. IEEE 802.11b/a/g differ in their physical layers, but share the same MAC layer specification (with minor variations). IEEE 802.11b operates at the 2.4GHz RF range and deploys DSSS for its most robust mode. IEEE 802.11a operates at the 5GHz RF range and deploys *Orthogonal Frequency Division Multiplexing* (OFDM) [49–51] in physical layer. IEEE 802.11g is basically the union of 802.11b and 802.11a. At MAC layer, IEEE 802.11 operates under *Distributed Coordination Function* (DCF) paradigm, which carries out CSMA/CA and MACAW [52] MAC protocols. DCF is therefore contention/random-backoff based and is not for real-time communications. In contrast, IEEE 802.11 also specifies the *Point Coordination Function* (PCF) paradigm, where the base station polls each remote station. PCF is contention-free and is

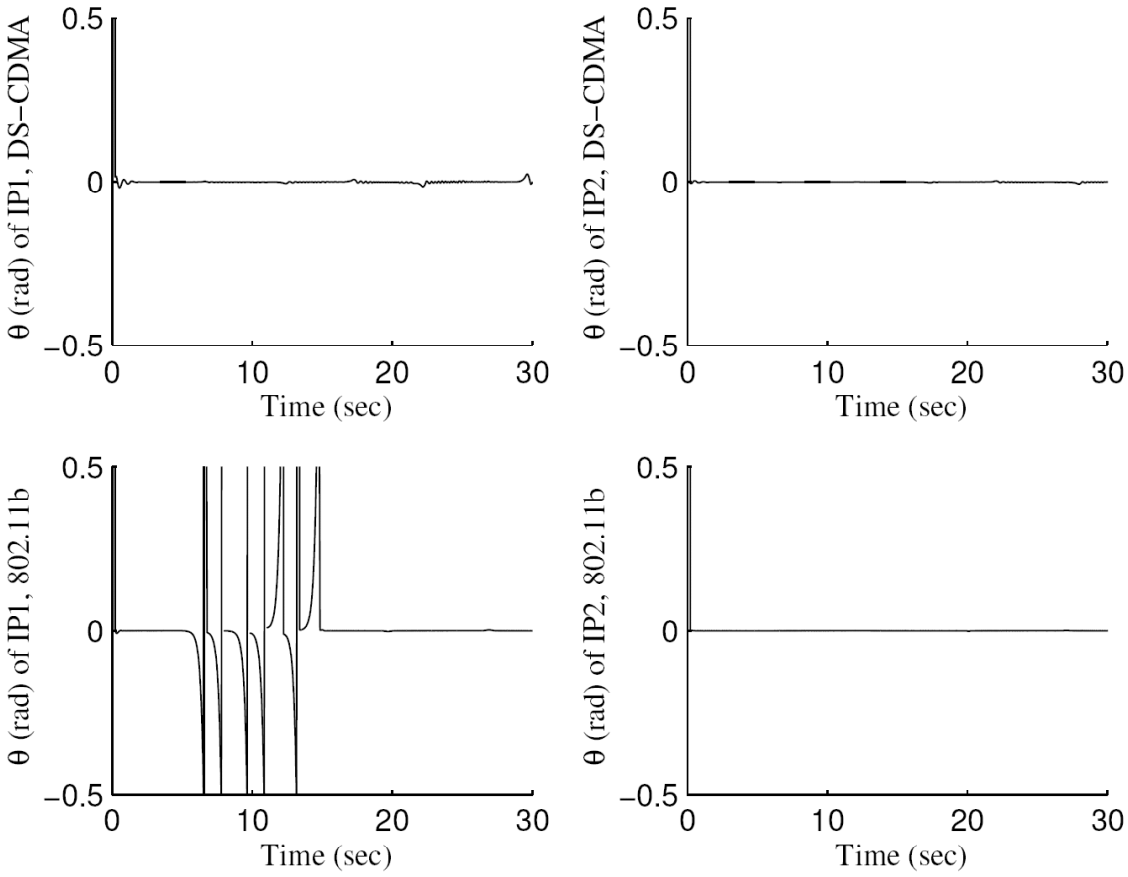


Figure 2.3: Simulation Results (θ traces)

hence for real-time communications.

IEEE 802.11 was designed for on demand high data rate bursty communications in office/home applications, such as FTP, emails and Web browsing. This mismatches the needs of most RT-WLAN control loops, where the demand for data throughput is low (typical packet lengths are $100 \sim 200$ bits, and minimal acceptable sampling/actuating rates are $1 \sim 10$Hz, or lower), while the demand for robustness is high: sampling/actuating packets must be delivered in real-time even under adverse channel conditions. In the following, we see IEEE 802.11's tolerance of adverse channel conditions is much inferior to that of the proposed DSSS-CDMA scheme, which fully exploits the low data throughput feature of RT-WLAN.

The comparisons between DSSS-CDMA and IEEE 802.11 are based on Monte Carlo simulations. In the simulator, the industrial indoor environment is a square room of 20m $\times$ 20m. The base station sits in the center while N remote stations scatter across the room according to uniform random distribution. Each remote station corresponds to a wireless control loop. The value of N varies from 1 to 100. Without loss of generality, the size of all sampling/actuating packets is 152 bits (the same as the inverted pendulum case,

Table 2.2: Phy. Settings for DSSS-CDMA/IEEE802.11 Comparisons

	Max per node trans power*	RF mainlobe bandwidth†‡
DSSS-CDMA vs. IEEE 802.11b	1watt	22MHz
DSSS-CDMA vs. IEEE 802.11a	800mw	18MHz for IEEE 802.11a, and 14.6MHz for DSSS-CDMA‡

∗ According to FCC regulation.

† Mainlobe is the main part of a signal's RF spectrum that carries information.

For DBPSK, BPSK and QPSK RF modulation that are used in IEEE 802.11b, IEEE 802.15.4i/ii and our proposed DSSS-CDMA scheme, the RF mainlobe bandwidth is equivalent to two times of chip rate. IEEE 802.15.4iii uses O-QPSK, where the RF mainlobe bandwidth is equivalent to the chip rate. The mainlobe bandwidth of IEEE 802.11a 6Mbps mode (the most robust mode of IEEE 802.11a) can be regarded as 18MHz, although IEEE 802.11a has a special RF spectrum shape due to OFDM.

Note: in this paper, when referring to DSSS-CDMA cellphone scheme, the mainlobe counts both the downlink and uplink RF spectra.

‡ According to IEEE 802.11 specifications, one single IEEE 802.11b RF channel has an RF mainlobe bandwidth of 22MHz, one single IEEE 802.11a RF channel has an RF mainlobe bandwidth of 18MHz. Due to difference between OFDM and DSSS modulations, letting DSSS-CDMA have an RF mainlobe bandwidth of 14.6MHz in the DSSS-CDMA versus IEEE 802.11a comparison safely makes the comparison pessimistic on the DSSS-CDMA side.

a typical control packet size), and all control loops have the same minimal acceptable sampling/actuating rate f^{min}. Two values of f^{min} are tested: 1Hz and 10Hz, which are typical for distributed industrial control loops. Every sampling/actuating packet must be delivered with success probability of no less than 0.999, that is, the maximal acceptable packet error rate Θ^{per} is 0.001. For a given N, f^{min} and RT-WLAN scheme (DSSS-CDMA, IEEE 802.11b, or IEEE 802.11a), 200 trials are simulated. In each trial, an instance of remote station layout and an instance of the wireless medium are generated (the wireless medium instance follows the random model depicted in Table 2.1). Each trial calculates its J^{min}: the minimal external RF interference power needed to disrupt at least one wireless control loop (see (2.25)). J^{min} is the quantitative robustness indicator compared between the DSSS-CDMA and IEEE 802.11b/a RT-WLAN schemes.

To make fair comparisons, parameters relevant to both schemes are set according to Table 2.2. Scheme-specific details are as follows:

For IEEE 802.11b, the most robust 1Mbps mode is deployed. For IEEE 802.11a, the most robust 6Mbps mode is deployed, *Note for IEEE 802.11b/a schemes, the packet is retransmitted as many times as possible throughout the sampling/actuating period, so as to increase the chance of successful delivery.*

For DSSS-CDMA, QPSK RF modulation/demodulation is deployed, with the RF band evenly divided into two halves: one for downlink and the other for uplink. The DSSS-CDMA scheme also deploys per-node pilot tone. The pilot tone is allocated with the same transmission power as any outgoing data channel of the node. To achieve maximal robustness, we shall set processing gain g_n according to Proposition 2.2.1. As-

suming the hardware-dependent upper bound on processing gain g^{max} is sufficiently large[3], Proposition 2.2.1 implies that the processing gain g_n for control loop n shall be $\lfloor r_c/(f_n^{min} L_n^{pkt}) \rfloor$. Specifically, given packet bit length ($L_n^{pkt} = 152$bit) and the RF mainlobe bandwidth (which decides chip rate r_c, see Table 2.2 footnote † for further explanation) listed in Table 2.2, when $f^{min} = 1$Hz and 10Hz, the corresponding processing gains are 36184 and 3618 for DSSS-CDMA/IEEE 802.11b comparison, and 24013, 2401 for DSSS-CDMA/IEEE 802.11a comparison[4].

The calculation of J^{min} for DSSS-CDMA and IEEE 802.11b/a schemes are based on their respective PER(BER)-SNR relationships. For DSSS-CDMA, the upper bound of BER under specified SNR is given in inequality (2.1). For IEEE 802.11b 1Mbps mode, inequality (2.26) gives the lower bound of BER under given SNR [39]:

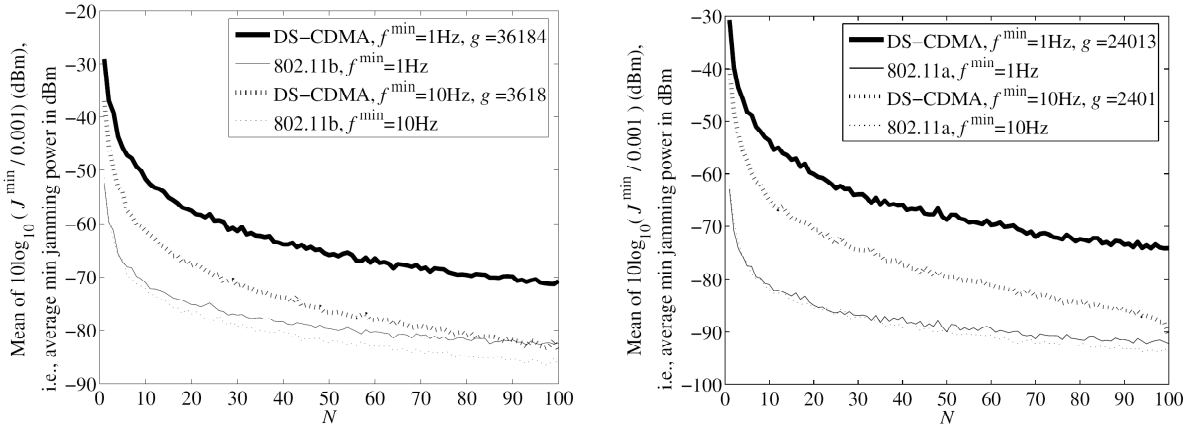
$$\mathcal{P}_{ber}^{80211b} \geq \frac{1}{2} \operatorname{erfc} \sqrt{\frac{gP_u}{J}}, \tag{2.26}$$

where g is the processing gain, P_u is the received signal power, J is the received total external RF interference power, and erfc is the well-known *complementary error function* [39]. The IEEE 802.11a 6Mbps mode deploys BPSK and 1/2 convolutional code for error correction. The corresponding PER-SNR relationship can be empirically derived through Monte Carlo simulations. Based on these BER(PER)-SNR relationships, J^{min}s of DSSS-CDMA and IEEE 802.11b/a schemes can be calculated. Fig. 2.4(a) and (b) compare these J^{min}s derived in all Monte Carlo trials. This comparison is pessimistic on the DSSS-CDMA side and optimistic on the IEEE 802.11b/a side because of many reasons: i) the *upper bound* of BER is used for DSSS-CDMA scheme, while for IEEE 802.11b/a the *lower bound* of BER and empirical *exact* PER are used respectively. ii) in inequality (2.1), the intended signal power P_u is included as part of interference to provide a (overly) pessimistic estimation on ISI; while for IEEE 802.11b/a, ISI is assumed to be 0. Therefore, in Fig. 2.4(a) and (b), the curves for DSSS-CDMA are J^{min} lower bounds while the curves for IEEE 802.11b/a are J^{min} upper bounds.

According to Fig. 2.4, the proposed DSSS-CDMA scheme can tolerate much bigger external RF interference power than the corresponding IEEE 802.11 schemes. When $f^{min} = 10$Hz and 1Hz, DSSS-CDMA achieves approximately 10dB and 20dB improvements on robustness than IEEE 802.11b, and approximately 20dB and 30dB improvements than IEEE 802.11a respectively. This is because the DSSS-CDMA scheme fully exploits the low data rate feature of industrial control loops by setting processing gain according to Proposition 2.2.1. When the data rate demand of control loop decreases (i.e. with smaller f^{min}), larger

[3]The upper bound on processing gain can increase exponentially when hardware increases. For example, with 61 registers, it is enough to produce PN sequence of $2,305,843,009,213,693,951$ chips, which is enough to allow any processing gain in practice [38].

[4]Note RF mainlobe bandwidth determines chip rate r_c. For a given chip rate r_c, any processing gain g can be picked, but a bigger g corresponds to a slower bit rate $r_b = r_c/g$.



(a) Comparison with IEEE 802.11b (b) Comparison with IEEE 802.11a

Figure 2.4: Robustness comparison between DSSS-CDMA and IEEE 802.11b/a RT-WLANs. J^{min}(watt) is the minimal external RF interference power needed to disrupt at least one wireless control loop (note P(watt) equals $10\log_{10}(P/0.001)$(dBm)). N is the number of wireless control loops. Note the curves for DSSS-CDMA are lower bounds for J^{min}, while the curves for IEEE 802.11b/a are upper bounds.

processing gain can be deployed and the corresponding tolerable external RF interference power increases.

2.3.3 Comparisons to IEEE 802.15.4

IEEE 802.15.4 [35] is a PHY/MAC standard for low data rate *Wireless Personal Area Networks* (WPAN). Recently, however, there is growing interest in applying IEEE 802.15.4 to *ad hoc* wireless sensor networks in efforts such as ZigBee [33]. Similar to IEEE 802.11 DCF and PCF, IEEE 802.15.4 also has two paradigms: *Contention Based* (CB) and *Contention Free* (CF). IEEE 802.15.4 CB mode uses CSMA/CA MAC, which is not for real-time communications. IEEE 802.15.4 CF mode is a centralized polling scheme (almost the same as IEEE 802.11 PCF), which supports real-time communications. Therefore, we compare DSSS-CDMA scheme with IEEE 802.15.4 CF.

Similar to IEEE 802.11, IEEE 802.15.4 can be further categorized into IEEE 802.15.4i/ii/iii according to their assigned RF ranges (see Table 2.3).

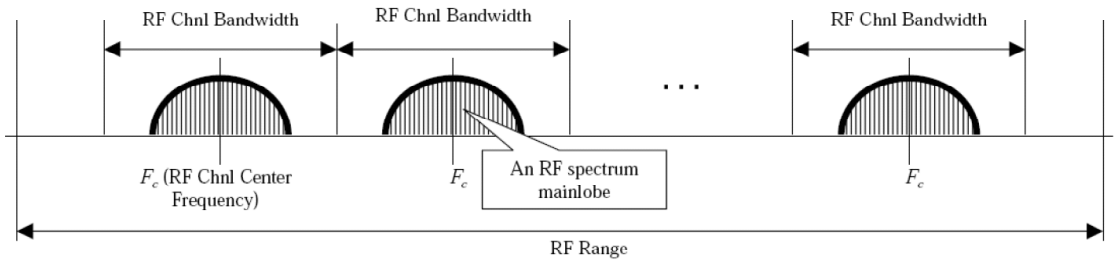


Figure 2.5: Definitions of RF range, RF channel, and RF channel bandwidth

Table 2.3: IEEE 802.15.4 Subtypes

Subtype	RF range* (MHz)	Per RF chnl bandwidth* (MHz)	Number of RF chnls*	Max per node trans power (Watt)
i†	868 ∼ 868.6	0.6	1	0.025†
ii	902 ∼ 928	2	10	1‡
iii	2400 ∼ 2483.5	5	16	1‡

Subtype	Chip rate (kcps)	RF mainlobe bandwidth (MHz)	Modulation	Bit rate (kbps)	Symbol rate (ksps)	Symbols
i†	300	0.6	BPSK	20	20	Binary
ii	600	1.2	BPSK	40	40	Binary
iii	2000	2	O-QPSK	250	62.5	16-ary Orthogonal

∗ See Fig. 2.5 for definitions of "RF range", "RF chnl", "Per RF chnl bandwidth", and "Number of RF chnls".
† Only allowed in Europe.
‡ According to FCC.

We carry out the same Monte Carlo simulation as Section 2.3.2 to compare DSSS-CDMA and IEEE 802.15.4i/ii/iii, using the wireless medium model given in Table 2.1 and the common physical layer settings given in Table 2.4.

The scheme-dependent details of DSSS-CDMA are slightly different from that of Section 2.3.2: when $f^{min} = 1$Hz and 10Hz, the corresponding processing gains are 987 and 99 for DSSS-CDMA/IEEE 802.15.4i comparison; 1974 and 197 for DSSS-CDMA/IEEE 802.15.4ii comparison; and 3289, 329 for DSSS-CDMA/IEEE 802.15.4iii comparison. These differences are due to change of RF mainlobe bandwidths in our comparisons.

Note for IEEE 802.15.4i/ii/iii schemes, the packet is retransmitted as many times as possible throughout the sampling/actuating period, so as to increase the chance of successful delivery.

We still use J^{min}, the minimal external RF interference power needed to disrupt at least one control loop, as the indicator of RT-WLAN robustness. The calculation of J^{min} for DSSS-CDMA and IEEE 802.15.4i/ii/iii schemes are still based on BER-SNR relationships ("BER" for *Bit Error Rate*). The DSSS-CDMA BER-SNR relationship is still given in inequality (2.1). For IEEE 802.15.4i/ii, since they both use BPSK (see Table 2.3), their BER-SNR relationship still follows inequality (2.26). IEEE 802.15.4iii, however, uses O-QPSK RF modulation/demodulation with 32-chip pseudo-orthogonal coding (see Table 2.3 and [35]). Such scheme makes it hard to derive a closed-form tight lower bound on BER for given SNR. Fortunately, Monte Carlo simulation can still give an empirical BER lower bound that is tight enough. Based on the

Table 2.4: Phy. Settings for DSSS-CDMA/IEEE802.15.4 Comparisons

	Max per node trans power*	RF mainlobe bandwidth†
DSSS-CDMA vs. IEEE 802.15.4i	25mw	0.6MHz
DSSS-CDMA vs. IEEE 802.15.4ii	1watt	1.2MHz
DSSS-CDMA vs. IEEE 802.15.4iii	1watt	2MHz

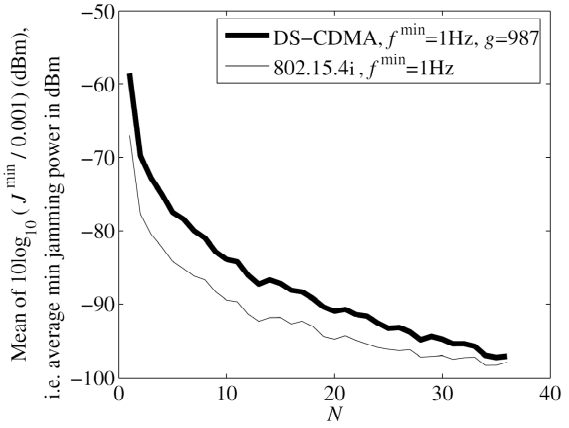
$*$ According to European regulation for IEEE 802.15.4i (FCC forbids free usage of 868 $\sim$ 868.6MHz for IEEE 802.15.4i) and FCC regulation for IEEE 802.15.4ii/iii.
† According to IEEE 802.15.4 specification. Also see Table 2.2 footnote † for definitions and discussions on mainlobe.

above BER-SNR relationships, J^{min} of DSSS-CDMA and IEEE 802.15.4i/ii/iii schemes can be calculated. Fig. 2.6(a), (b) and (c) compare these J^{min}s derived in Monte Carlo simulations. As stated in Section 2.3.2, the comparisons are still pessimistic on the DSSS-CDMA side and optimistic on the IEEE 802.15.4i/ii/iii side. That is, in Fig. 2.6(a), (b) and (c), the curves for DSSS-CDMA are J^{min} lower bounds, and the curves for IEEE 802.15.4 are J^{min} upper bounds.

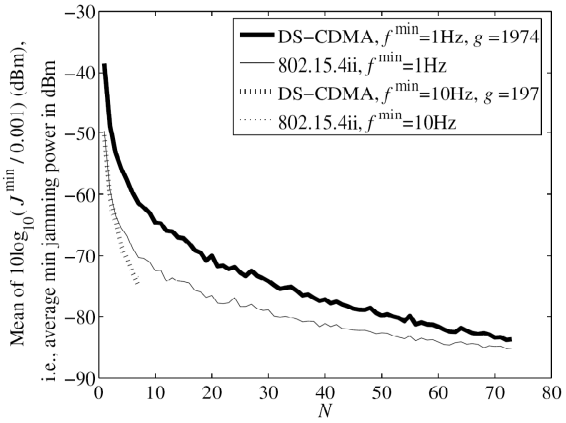
According to Fig. 2.6, when the sampling/actuating rate is low (see the $f^{min} = 1$Hz curves), DSSS-CDMA significantly out-performs IEEE 802.15.4i/ii/iii on robustness. When the sampling/actuating rate is high, however, DSSS-CDMA only performs better when N (total number of control loops) is small, and becomes inferior to IEEE 802.15.4ii/iii when N is large enough (see the $f^{min} = 10$Hz curves). This is because an IEEE 802.15.4i/ii/iii RF channel is of very narrow RF bandwidth. To squeeze into the same RF bandwidth, the chip rate of DSSS-CDMA scheme must be low. When the data throughput is high (since packet bit size is fixed to 152, a higher sampling/actuating rate f^{min} or a bigger N corresponds to a larger data throughput), DSSS-CDMA cannot deploy a basic-need processing gain g to overcome MAI.

Fortunately, the continuous RF bandwidth available is usually much wider than what is used by an IEEE 802.15.4 RF channel. For example, wherever an IEEE 802.11b WLAN can be deployed, the continuous RF bandwidth available is at least 22MHz, equivalent to 36.7, 11 and 4.4 times the RF bandwidth of an IEEE 802.15.4i, ii, and iii RF channel respectively. Such 22MHz RF bandwidth allows the DSSS-CDMA RT-WLAN mainlobe bandwidth to be 36.7, 18.3 and 11 times the IEEE 802.15.4i, ii, and iii mainlobe bandwidths respectively.

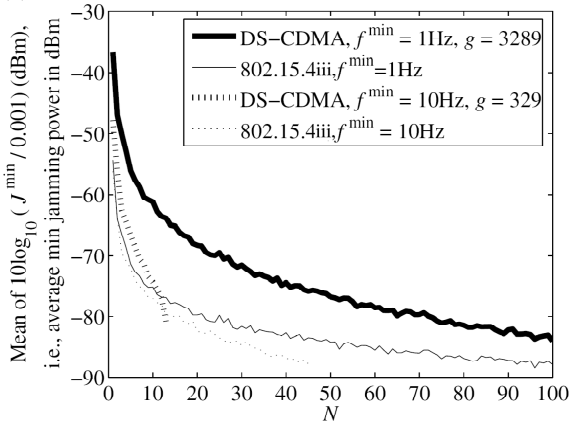
Another way of thinking is as follows: In a 3-D space, one can use 27 non-overlapping RF channels to color cells so that any two cells using the same RF channel are at least three hops (cells) away (see Fig. 2.7). The FCC *Industrial-Scientific-Medical* (ISM) RF ranges allow 27 non-overlapping RF channels, each with a continuous RF bandwidth of at least 14MHz, which is 23.3, 7 and 2.8 times the RF bandwidth of an IEEE 802.15.4i, ii, and iii RF channel respectively. In terms of RF mainlobes, such 14MHz RF bandwidth allows



(a) Comparison with IEEE 802.15.4i



(b) Comparison with IEEE 802.15.4ii



(c) Comparison with IEEE 802.15.4iii

Figure 2.6: Robustness comparison between DSSS-CDMA and IEEE 802.15.4i/ii/iii RT-WLANs. The meaning of J^{min} and "dBm" is the same as in Fig. 2.4. N is the number of wireless control loops. The curves for DSSS-CDMA are lower bounds for J^{min}, while the curves for IEEE 802.15.4i/ii/iii are upper bounds. Note for DSSS-CDMA/IEEE 802.15.4i comparison, when $f^{min} = 10$Hz, with a packet size of 152 bits, an IEEE 802.15.4i RT-WLAN can only afford 3 control loops. Such RT-WLAN is practically useless. Similarly, when $f^{min} = 10$Hz, an IEEE 802.15.4ii/iii RT-WLAN can only afford 7 and 45 control loops respectively; when $f^{min} = 1$Hz, an IEEE 802.15.4ii RT-WLAN can only afford 73 control loops.

the DSSS-CDMA RT-WLAN mainlobe bandwidth to be 23.3, 11.7 and 7 times the IEEE 802.15.4i, ii, and iii mainlobe bandwidths respectively.

Therefore, the comparisons in Fig. 2.6 are too pessimistic on the DSSS-CDMA side. Given a layout of IEEE 802.15.4i/ii/iii base stations, for each corresponding RT-WLAN, the continuous RF bandwidth available is usually much bigger than that of an IEEE 802.15.4i/ii/iii RF channel. If DSSS-CDMA fully utilizes the RF bandwidth available, the DSSS-CDMA performance can improve significantly.

In the following, we redo the Monte Carlo comparisons between DSSS-CDMA and IEEE 802.15.4i/ii/iii with modified physical layer settings as shown in Table 2.5. According to Table 2.5, the RF mainlobe bandwidth of DSSS-CDMA is w times the mainlobe bandwidth of an IEEE 802.15.4i/ii/iii RF channel.

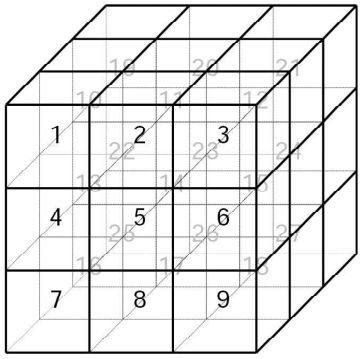


Figure 2.7: Twenty-seven colors are enough to color the cells in a 3-D space so that any two same-color cells are at least 3 hops (cells) away.

Table 2.5: Phy. Settings for DSSS-CDMA/IEEE802.15.4 Comparisons

	Max per node trans power*	RF mainlobe bandwidth†
DSSS-CDMA vs. IEEE 802.15.4i	25mw	600kHz for IEEE 802.15.4i, $w \times 600$kHz for DSSS-CDMA
DSSS-CDMA vs. IEEE 802.15.4ii	1watt	1.2MHz for IEEE 802.15.4ii, $w \times 1.2$MHz for DSSS-CDMA
DSSS-CDMA vs. IEEE 802.15.4iii	1watt	2MHz for IEEE 802.15.4iii, $w \times 2$MHz for DSSS-CDMA

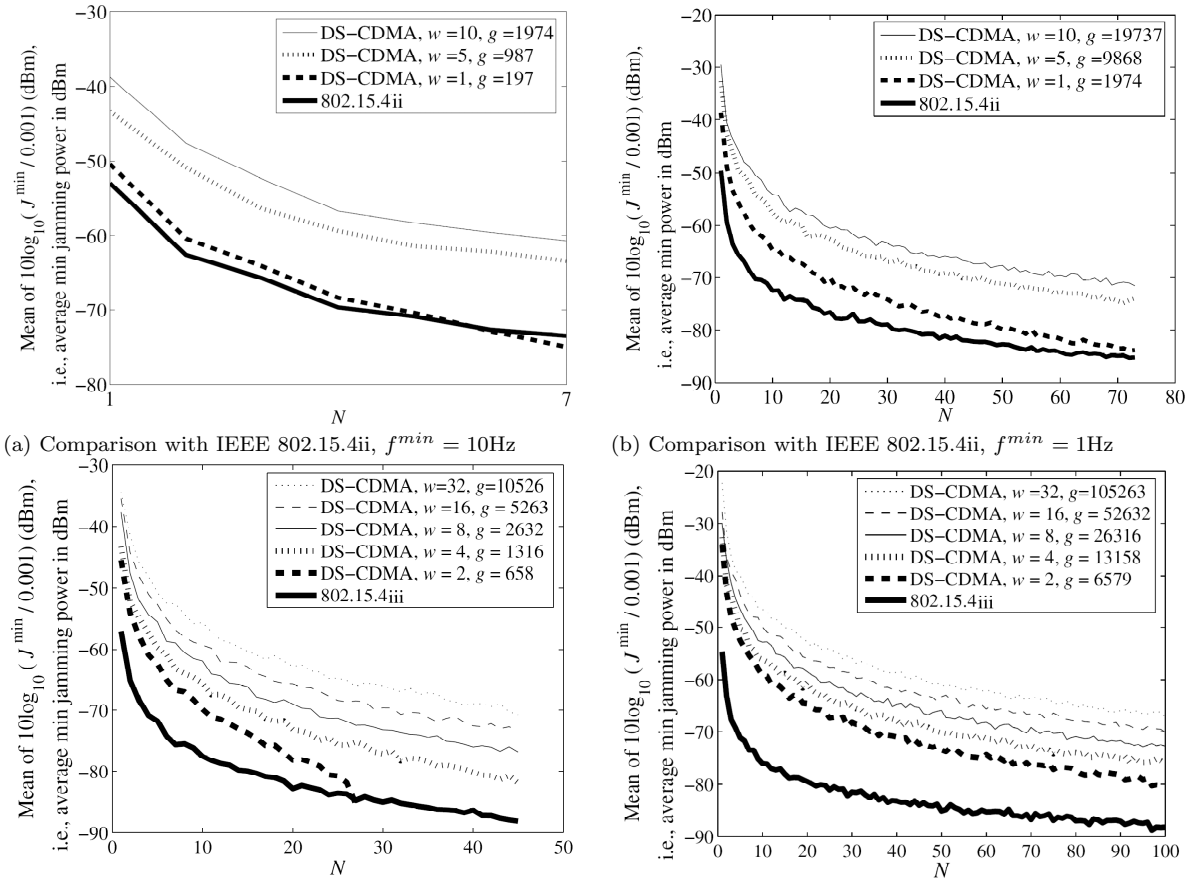
$*$ According to European regulation for IEEE 802.15.4i and FCC regulation for IEEE 802.15.4ii/iii.
† According to IEEE 802.15.4 specification. Also see Table 2.2 footnote † for definitions and discussions on mainlobe.

Different w's are evaluated, with the results plotted in Fig. 2.8.

Note in Fig. 2.8, we do not compare DSSS-CDMA with IEEE 802.15.4i. This is because: i) IEEE 802.15.4i uses the same modulation/demodulation scheme as IEEE 802.15.4ii; ii) IEEE 802.15.4i allows much lower maximal transmission power (0.025Watt) than IEEE 802.15.4ii (1Watt); iii) IEEE 802.15.4i provides half the bit rate (and chip rate) of IEEE 802.15.4ii; iv) IEEE 802.15.4i is not allowed by FCC (only allowed in Europe). Therefore, in terms of either robustness or data throughput, IEEE 802.15.4i is inferior to IEEE 802.15.4ii. Given the DSSS-CDMA versus IEEE 802.15.4ii comparisons, comparisons to IEEE 802.15.4i are redundant.

2.3.4 Discussion on Error Correction Coding

Another way to exploit low data rate for higher robustness is dedicating the redundant bandwidth to error correction code. The most popular error correction coding is the convolutional coding. At the sender end, a convolutional encoder (k, n, m) encodes every k input bits into n output bits using m shift memory registers,



(a) Comparison with IEEE 802.15.4ii, $f^{min} = 10$Hz

(b) Comparison with IEEE 802.15.4ii, $f^{min} = 1$Hz

(c) Comparison with IEEE 802.15.4iii, $f^{min} = 10$Hz

(d) Comparison with IEEE 802.15.4iii, $f^{min} = 1$Hz

Figure 2.8: Robustness comparison between DSSS-CDMA and IEEE 802.15.4ii/iii RT-WLANs, when the DSSS-CDMA RF mainlobe bandwidth is w times the mainlobe bandwidth of an IEEE 802.15.4ii/iii RF channel (see Table 2.5). J^{min}(watt) is the minimal external RF interference power needed to disrupt at least one wireless control loop. N is the number of wireless control loops. Note the curves for DSSS-CDMA are lower bounds for J^{min} while the curves for IEEE 802.15.4ii/iii are upper bounds. Also note when $f^{min} = 10$Hz, an IEEE 802.15.4ii/iii RT-WLAN can only afford 7 and 45 control loops respectively; when $f^{min} = 1$Hz, an IEEE 802.15.4ii RT-WLAN can only afford 73 control loops.

where

$$m \geq \lceil \log_2 n \rceil \tag{2.27}$$

to produce practical convolutional codes. Such encoder corresponds to a coding rate $R \stackrel{def}{=} \frac{k}{n}$. The upper bound of coding gain is decided by k, n and m. Assuming binary antipodal symbol signal and AWGN channel, convolutional coding achieves a gain of Rd_f – which is often called the *Asymptotic Coding Gain* (ACG) – on SNR, where R is the coding rate and d_f is the free distance of convolutional codes. Denote ACG as g_{acg}. A loose upper bound of d_f is $n(m+1)$, so ACG g_{acg} is upper bounded by:

$$g_{acg} = Rd_f \leq Rn(m+1) = k(m+1). \tag{2.28}$$

Suppose the available wireless medium bandwidth is B_{medium}(bps), and the information bandwidth is B_{info}(bps).

Using DSSS, the maximal gain on SNR is a processing gain

$$g = B_{medium}/B_{info} \qquad \text{(according to Proposition 2.2.1).} \tag{2.29}$$

Using convolutional coding, the maximal gain on SNR is an ACG of $g_{acg} = k(m+1)$. To achieve the maximal ACG, all redundant bandwidth shall be dedicated to convolutional coding, that is, coding rate $R = B_{info}/B_{medium}$, in other words, $n = kB_{medium}/B_{info}$. Because of inequality (2.27), we shall pick $m = \lceil \log_2 n \rceil = \lceil \log_2(kB_{medium}/B_{info}) \rceil$ (why not to pick a bigger m is explained later). Therefore, the maximal gain on SNR using convolutional coding is:

$$g_{acg} = k(m+1) = k\left(\left\lceil \log_2 \frac{kB_{medium}}{B_{info}} \right\rceil + 1\right). \tag{2.30}$$

For the typical RT-WLAN scenario where B_{medium} equals 10Mbps and B_{info} varies from 1bps to 100Kbps, Fig. 2.9 compares the SNR gain between using DSSS and using convolutional coding with $k = 1$ and $m = \lceil \log_2 n \rceil = \lceil \log_2(kB_{medium}/B_{info}) \rceil$. According to Fig. 2.9, DSSS significantly outperforms convolutional coding on improving SNR.

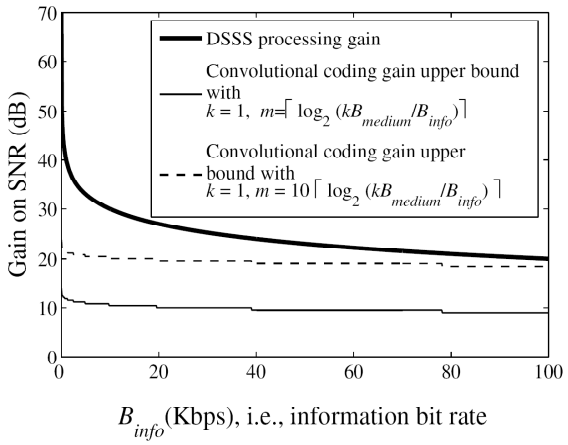


Figure 2.9: Comparison on gain over SNR. The available wireless medium bandwidth $B_{medium} = 10$Mbps; information data bit rate B_{info} varies from 1bps to 100Kbps. All redundant bandwidth are dedicated to achieving higher gain on SNR: in the DSSS scheme, $g = B_{medium}/B_{info}$; in the convolutional coding scheme, $n = kB_{medium}/B_{info}$ (note convolutional coding gain (ACG g_{acg}) is upper bounded by $k(m+1)$, where m must be no less than $\lceil \log_2 n \rceil$).

Because $g_{acg} \leq k(m+1)$, some may argue if a bigger k or m is picked, the performance of ACG may

be better. Nevertheless, even with the small value of $k = 1$ and $m = \lceil \log_2 n \rceil = \lceil \log_2(kB_{medium}/B_{info}) \rceil$, the convolutional coding used in Fig. 2.9 is already impractical: Empirically, no good convolutional coding scheme with $n > 99$ is known. To achieve maximal ACG, however, the entire bandwidth is dedicated for convolutional code, that is, $n = kB_{medium}/B_{info}$. When $k = 1$, $B_{medium} = 10$Mbps, and B_{info} varies from 1bps to 100Kbps, n varies from 10^7 to 100, all exceeding 99. Picking bigger k or m does not help solve this problem.

Even if convolutional coding schemes with $n > 99$ are found, decoder complexity may still prevent us from picking a bigger k or m: For a convolutional decoder, the number of algorithmic operations per second (denoted as Op) is [53]:

$$Op = c2^{km+1}B_{info}/k, \tag{2.31}$$

where c is an implementation dependent positive constant, empirically no less than 1. Fig. 2.10 plots Op when $k = 2$, $m = \lceil \log_2 n \rceil = \lceil \log_2(kB_{medium}/B_{info}) \rceil$, $c = 1$, and $B_{medium} = 10$Mbps. The decoding complexity is daunting ($Op \approx 10^{10}$). Since picking a bigger k is infeasible, the alternative is to pick a bigger m. According to Fig. 2.9, if $k = 1$, m must be at least $10\lceil \log_2 n \rceil = 10\lceil \log_2(kB_{medium}/B_{info}) \rceil$ to let convolutional coding ACG outperform DSSS processing gain. Fig. 2.10 also plots Op for this case, that is, $k = 1$, $m = 10\lceil \log_2 n \rceil = 10\lceil \log_2(kB_{medium}/B_{info}) \rceil$, $c = 1$, and $B_{medium} = 10$Mbps. The decoding complexity is even more daunting ($Op > 10^{20}$). Therefore, picking a bigger k or m is not the way out for convolutional coding to outperform DSSS.

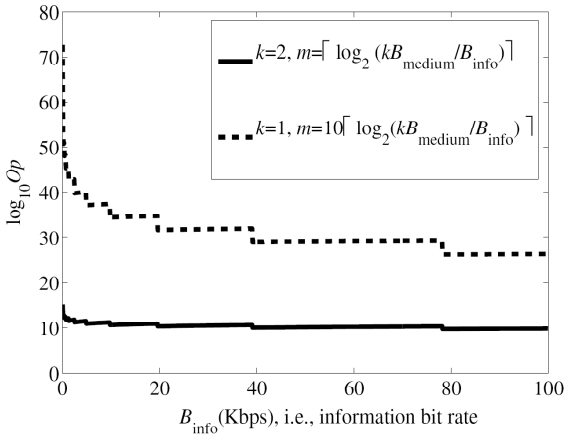


Figure 2.10: Convolutional Decoding Complexity

2.4 Related Work

One purpose of this chapter is to demonstrate DSSS-CDMA cellphone network paradigm is more appropriate for RT-WLAN than the dominant IEEE 802.11/802.15.4 WLAN paradigms. Intuitively, industrial control loop and cellphone voice session bear many similarities: they are both low-data-rate regular traffic, and last for long duration in a session-like pattern. The main difference lies in the high robustness concern for industrial control loops. Such concern calls for better exploitation of the low data rate feature to provide more robustness. Current CDMA cellphone network architectures have not yet focused on such demand. However, the current CDMA cellphone network architectures [45, 54–56] provide good foundations to start building our proposed DSSS-CDMA RT-WLAN with. The technologies needed by our scheme are already mature, specifically, the capability of providing multiple reconfigurable CDMA channels, processing gain options and power levels are already standard practices supported by most contemporary CDMA cellphone chip sets, such as QualComm CSM6800, CSM6700, CSM5500 [57, 58] etc. The major modification pending is to better customize the configurable options and the resource management strategies according to the industrial control needs.

We have shown that DSSS-CDMA RT-WLAN can achieve much higher robustness than IEEE 802.11 WLANs [34,46–48] and IEEE 802.15.4 WLANs [35], both of which have fixed robustness levels. Nonetheless, if application-dependent processing gain configuration is provided for IEEE 802.11b or IEEE 802.15.4, its robustness can also be greatly improved. This is exactly the DSSS-*TDMA* RT-WLAN approach, which is shown to be less preferable than DSSS-CDMA for three weaknesses (see Section 2.1). However, it still merits further study on how to overcome these three weaknesses so as to make DSSS-TDMA RT-WLAN a feasible and competitive scheme.

In addition to IEEE 802.15.4, we have another WPAN MAC/physical layer standard IEEE 802.15.1 [59], a.k.a. Bluetooth, which is mainly designed for high-data-rate, low-power and short-range communications between PC and its peripherals. Bluetooth is known to have robustness inferior to IEEE 802.11b [60].

Also, at the physical layer, *Frequency Hopping Spread Spectrum* (FHSS) [25] and DSSS bear great resemblance. That is why under many circumstances, FHSS and DSSS are interchangeable. However, FHSS is less desirable than DSSS in hardware cost and system complexity. And digital wireless FHSS systems that carries out FHSS within every bit duration (so as to achieve processing gain) are not as widely available as DSSS systems.

Finally, our goal is noticeably to maintain wireless control loop communications under channel conditions as harsh as possible, rather than to make wireless control loop communications immune to adverse channel conditions.

The content of this chapter is published in [15] and [61].

2.5 Conclusion

The top priority for building *Real-Time Wireless LAN* (RT-WLAN) is robustness: wireless control loops must be maintained under all adverse channel conditions. Specifically, power attenuation may change drastically because of large-scale path loss and fading; contending RF devices may be turned on accidentally or maliciously; for industrial environments, the situation is even worse because of various EMI sources such as electric motor and welding, and serious large-scale-path-loss/fading due to heavy obstructions. The robustness requirement makes the IEEE 802.11 WLANs (mainly designed for irregular office/home data traffics) inappropriate for RT-WLAN. In contrast, industrial control loop traffics are mostly regular sustained traffics with extremely low data rates. This feature allows DSSS's deployment of high processing gain for high robustness.

According to fine-grained physical layer simulations and Monte Carlo comparisons, we show that by deploying slowest data bit rate (largest possible processing gain) allowed by minimal sampling/actuating rate, a DSSS RT-WLAN can provide significantly higher robustness than IEEE 802.11/802.15.4 WLAN. At the MAC layer, although either CDMA or TDMA can be deployed, CDMA is more preferable than TDMA for its ease of scheduling, overrun isolation and low overhead for regular sustained traffics. Therefore, we claim that by fully exploiting low-data-rate feature of industrial control loops, DSSS-CDMA better meets the needs of RT-WLAN. That is, we open a new application domain where the CDMA cellphone network paradigm would prevail again due to its unique characteristics. Though some modifications are needed, it is promising to build our proposed DSSS-CDMA RT-WLAN scheme on top of the many contemporary CDMA cellphone network architectures.

DSSS-CDMA RT-WLAN scheme opens a new problem space: many variables can be configured, such as processing gain, data rate, transmission power, number of channels per control loops and acceptable packet error rate threshold; and many objectives can be pursued, such as efficient planning algorithms, capacity, utility, and coexistence of regular-low-throughput versus irregular-high-throughput traffics. Also, the situation will be more complicated for multiple cells. We are interested in carrying out further studies in all these directions.

Chapter 3

REAL-TIME SWITCH

We know that to support the many CPS applications, such as telepresence, next generation industrial control, next generation vehicle, and MDPnP, a wired real-time *Wide Area Network* (WAN) is indispensable. A wired real-time WAN can serve as a reliable backbone, provide large data throughput, and reuse legacy facilities.

To build such WANs, we must first have real-time switch (router). However, nowadays commercially available switches are not real-time, they are instead tailored towards best-effort Internet traffic. Therefore, we must start from designing a real-time switch.

There are three approaches to build a switch: output queueing, input queueing, and *Virtual Output Queueing* (VOQ).

In output queueing, queueing only takes place at the output ports (simplified as *outputs* in the following). When a packet arrives at an input port (simplified as *inputs* in the following), it immediately goes to the queue at its destined output. Due to its simplicity, most QoS scheduling algorithms, such as WFQ [62], WF^2Q [63], Deficit Round-Robin [64] etc., assumes output queueing [65].

Output queueing, however, creates a data bus bottleneck. Since there is no queue at the inputs, the data bus must deliver every arriving packet to output queue immediately. In the worst case, every input may reach its maximum capacity, and all incoming packets may go to a same output. Therefore, the data bus connected to each output must provide a capacity no less than total capacity of all inputs. Suppose a switch has N inputs, each with a data line rate of C, then the data bus connected to each output must provide a capacity of $N \times C$. We call this N *speed-up*. Such a speed-up makes output queueing undesirable for high-speed switches or switches with large number of ports (N) because of the challenges of developing high-speed memory banks.

In contrast to output queueing, input queueing buffers packets in queues at the inputs. This avoids the need for speedup in the switch, but suffers from *head of line* (HOL) blocking: if packets going to other outputs are blocked at the head of the input queue, a packet to output j must wait for the depletion of this backlog before it is transferred to output j, even though output j is idle. It is well known that if each input queue is first-in-first-out (FIFO), HOL blocking can limit the throughput to just 58.6% [66].

The solution to the HOL problem is to deploy *virtual output queueing*, where each input maintains a virtual output queue for each output. VOQs eliminate HOL blocking, but packets from different inputs' VOQs still contend for the same output. Various schemes are proposed to reduce this contention, so as to improve the hardware utilization. To the best of our knowledge, the most popular scheme is *i*SLIP [13,67,68], which is elaborated upon in Section 3.1. Although *i*SLIP efficiently utilizes the switch hardware and is simple to implement, it does not provide real-time guarantees. In fact, real-time high-performance switch design is still an open problem [69].

In this article we describe a design of a real-time switch by making minimal modifications to *i*SLIP, or even by simplifying *i*SLIP. This design benefits switch manufacturers since *i*SLIP is already widely implemented in commercial products, and the minor modifications can be easily incorporated into the manufacturing process. Our approach is to define operations that allow a switch to serve each link l for C_l units of time every M units of time. This enables hierarchical scheduling, which can then be analyzed using one of several known techniques [70–72].

In the following, Section 3.1 describes the *i*SLIP scheme; Section 3.2 proposes our switch design for industrial real-time communications; Section 3.3 evaluates our design; Section 3.4 discusses related work; and Section 3.5 concludes the chapter.

3.1 Crossbar Switches and *i*SLIP

To support input queueing or VOQ, most high-performance switches use a crossbar hardware fabric [73] (Fig. 3.1). The data bus from each input (the horizontal line segments in the figure) intersects with the data bus of each output (the vertical line segments). The intersections can be turned on or off during runtime by the switch scheduling logic. To facilitate the scheduling logic, crossbar switches transfer packets in fixed-size fragments called *cells*; and the time to transfer one cell across the crossbar fabric is called a *cell-time*. Therefore, the scheduling logic works periodically: it determines a matching between inputs and outputs at the beginning of each cell-time; then all scheduled cells are transferred synchronously across the crossbar fabric, taking one cell-time; and then the next period starts, so on and so forth.

*i*SLIP [13, 67] is a popular scheduling mechanism for VOQ crossbar switches. Without loss of generality, suppose a switch consists of N inputs $I_1 \sim I_N$ and N outputs $O_1 \sim O_N$ (or an "$N \times N$ switch" in the following discussion). Under *i*SLIP, every input I_i maintains a circular list of outputs $O_1 \sim O_N$, with pointer a_i pointing to O_1 initially. This circular list is called the input's *round-robin schedule*. The output pointed to by a_i has the highest priority, the next output (modulo N) has the next highest priority, and so on. In

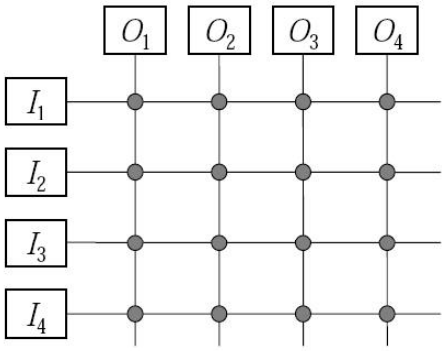


Figure 3.1: Crossbar Switch Hardware Fabric

the same way, every output O_j also maintains a round-robin schedule of inputs, with pointer g_j pointing to the highest priority input, the next input (modulo N) has the next highest priority, and so on.

With the above data structures, the basic iSLIP runs following steps [67]:

Step 1 *Request.* Each unmatched input sends a request to every output for which it has a queued cell.

Step 2 *Grant.* If an unmatched output receives any requests, it grants the requesting input with the highest priority in the output's round-robin schedule. The output notifies each input whether or not its request was granted. The pointer g_i to the round-robin schedule is incremented (modulo N) to one location beyond the granted input *if, and only if, the grant is accepted in Step 3.*

Step 3 *Accept.* If an input receives any grants, it accepts the granting output with the highest priority in the input's round-robin schedule. The pointer a_i to the round-robin schedule is incremented (modulo N) to one location beyond the accepted output.

Since some grants may not be accepted, iSLIP may carry out up to N iterations of Request-Grant-Accept at the beginning of each cell-time to increase the size of the matching.

The original iSLIP mechanism [13, 67] also accommodates several variations such as weighted iSLIP and prioritized iSLIP. Different commercial iSLIP switches may implement certain subsets of these variations. According to McKeown [13], iSLIP can achieve 100% throughput (i.e., every output reaches maximum capacity; in other words, the bipartite graph between inputs and outputs defined by the crossbar fabric reaches full match for every cell-time) for uniform traffic, and quickly adapts to a fair scheduling policy that never starve any input queue for non-uniform traffic.

However, obtaining accurate delay bounds for iSLIP is still an open problem. The best known iSLIP delay bound is still "very pessimistic" [69]. For example, if in an $N \times N$ iSLIP switch, every input has periodic real-time traffic going to every output, the known single hop delay bound for packets from input I_i

to output O_j is

$$d = N^2 \sum_k C_{ijk}, \tag{3.1}$$

where C_{ijk} is the per packet transmission time of the kth real-time flow going from I_i to O_j. Suppose $N = 32$, C_{ijk} is the same for all links and flows, and if there are 100 real-time flows going from I_i to O_i, then the single hop delay bound is at least 102400 times that of a packet transmission time.

3.2 A Real-Time Switch Design

To support real-time communication, we propose a real-time switch design by making minimum modifications to *i*SLIP. Interestingly, our design simplifies *i*SLIP rather than complicates it.

Firstly, we observe a large body of research on serving a real-time task or task-set with a real-time *virtual machine task* (VM-task) [65, 70–72, 74–76]. One simple and widely implemented form is clock-driven scheduling [65], where a VM-task (M, C) indicates that a real-time task or task-set is served C time units during each clock-period of M time units.

Using clock-driven scheduling, we may serve the kth real-time flow f_{ijk} from input I_i to output O_j in a crossbar switch with a VM-task (M, C_{ijk}) (*unless explicitly noted, the default time unit is "cell-time"*), where $k = 1, 2, \ldots, K_{ij}$, and K_{ij} is the total number of real-time flows going from I_i to O_j. That is, as long as the switch forwards C_{ijk} cells from I_i to O_j for f_{ijk} in each M cell-time clock-period, packets of f_{ijk} shall meet their local deadlines.

Secondly, *i*SLIP's request-grant-accept negotiation between inputs and outputs is for non-deterministic Internet traffic, which changes frequently. *If the traffic rarely changes and is periodic, as that of real-time flows in industrial networks, there is no need for a request-grant-accept negotiation.* Instead, deterministic grants (or accepts) alone suffice. We only need to work out a conflict-free grant (or accept) schedule during configuration-time.

In summary, our real-time switch shall serve each real-time flow with a real-time VM-task, and the VM-task is served with deterministic grant (or accept). We elaborate such design in the following.

3.2.1 Per-Flow VOQ

Our proposed real-time switch is an $N \times N$ crossbar VOQ switch. However, to control jitter for simple *end-to-end* (E2E) delay guarantee, we deploy *per-flow virtual output queueing* (per-flow VOQ), instead of combining all cells at input I_i destined for output O_j in one virtual output queue. In other words, if there

are K_{ij} flows going from I_i to O_j, then for O_j, we maintain K_{ij} queues at I_i for each flow respectively.

The overall buffer requirements at the switch do not change (much) because of the per-flow VOQs; the same packets that would have been buffered at one VOQ are held in different buffers depending on their flow id. Flow differentiation can be performed in conjunction with IP lookup and output port identification, therefore the hardware complexity and the per-cell processing time overhead increase only marginally. It is also worth mentioning that per-flow VOQs are simple FIFO queues. We do not need to maintain per-flow state information, or perform sorting (as most timestamp based QoS schemes, such as WFQ [62] and WF^{2}Q [63], do), which may affect performance.

3.2.2 Traffic Demand

All traffic demand in our real-time switch is abstracted by the clock-driven scheduling of VM-tasks (see Section 3.2.5 Equation (3.4)). According to clock-driven scheduling, the kth real-time flow f_{ijk} from I_i to O_j can be served by VM-task $\tau_{ijk} = (M, C_{ijk})$. That is, during each clock-period of M cell-time, C_{ijk} cells are forwarded from I_i to O_j for flow f_{ijk}.

Denote $C_{ij} \stackrel{def}{=} \sum_{k=1}^{K_{ij}} C_{ijk}$. That is, I_i needs to forward C_{ij} cells to O_j during each clock-period. Then the entire VM-task set $\{(M, C_{ijk})\}$ $(i = 1 \sim N, j = 1 \sim N, k = 1 \sim K_{ij})$ must meet the following constraints to be *feasible*:

Constraint 1 Feasible input utilization:

$$\sum_{j=1}^{N} C_{ij} \leq M, i = 1, 2, \ldots, N. \tag{3.2}$$

Constraint 2 Feasible output utilization:

$$\sum_{i=1}^{N} C_{ij} \leq M, j = 1, 2, \ldots, N. \tag{3.3}$$

Infeasible VM-task sets are unschedulable, and we do not consider them.

3.2.3 Runtime Scheduling

Corresponding to the M cell-time clock-period, each output O_j maintains a round-robin schedule S_j^{out} of M elements. The gth $(1 \leq g \leq M)$ element dictates the input from which O_j fetches a cell at the gth cell-time of a M cell-time clock-period. $S_1^{out} \sim S_N^{out}$ are *conflict-free*, meaning at any cell-time of the M cell-time clock-period, no two outputs fetch cells from the same input; and S_j^{out} $(j = 1 \sim N)$ has exactly C_{ij}

$(i = 1 \sim N)$ elements for input I_i, meaning O_j fetches C_{ij} cells from I_i in each M cell-time clock-period. We will describe how to derive $S_1^{out} \sim S_N^{out}$ in a later subsection (Section 3.2.4).

Correspondingly, each input I_i maintains a round-robin schedule S_{ij}^{in} of C_{ij} elements for each output O_j. The ath $(a = 1, 2, \ldots, C_{ij})$ element of S_{ij}^{in} indicates the per-flow VOQ to send a cell from, when I_i is to connect O_j for the ath time during the M cell-time clock-period. That is, S_{ij}^{in} has C_{ijk} elements for f_{ijk} $(k = 1 \sim K_{ij})$ respectively; and *these elements are arbitrarily ordered*.

Input I_i also maintains a pointer ρ_{ij} to S_{ij}^{in}, initially pointing to the first element of S_{ij}^{in}.

With the above settings, our proposed real-time switch only executes two steps at the beginning of the gth $(g = 1, 2, \ldots, M)$ cell-time of each M cell-time clock-period:

Step 1 *Grant.* Output O_j grants the input indicated by the gth element of S_j^{out}.

Step 2 *Accept.* On receiving a grant from O_j, input I_j sends O_j the head cell (or null if the queue is empty) of per-flow VOQ indicated by pointer ρ_{ij}. ρ_{ij} is increased by 1 (modulo C_{ij}).

The "Request" step in the original iSLIP disappears; and because $S_1^{out} \sim S_N^{out}$ are conflict-free, a "Grant" is always accepted , which eliminates the need of N iterations. Therefore, our real-time switch incurs $O(1)$ computation during runtime, and is simpler than iSLIP.

3.2.4 Configuration-Time Scheduling

During configuration-time, we need to work out conflict-free round-robin schedules $S_1^{out} \sim S_N^{out}$. In this section, we show that any feasible VM-task set has a conflict-free schedule that can be computed in polynomial time.

Theorem 3.2.1 *A VM-task set $\{(M, C_{ijk})\}$ has conflict-free schedules $S_1^{out} \sim S_N^{out}$ if and only if the VM-task set is feasible (see Constraint 1 and 2 for the definition of "feasible"); and any feasible VM-task set can be scheduled within $O(N^4)$ time, where N is the number of input (also output) ports.*

Proof: 1) Sufficiency: The scheduling of feasible VM-task set $\{(M, C_{ijk})\}$ can be reduced to a *preemptive open shop scheduling* (POSS) problem [77].

The preemptive open shop scheduling problem involves n tasks, denoted by the set $\{\tau_i\}$, and η machines $(n \geq 1, \eta \geq 1)$. τ_i has η subtasks, represented by the set $\{\tau_{ij}\}$, such that τ_{ij} has to be executed on machine j. Tasks can be preempted, and no restrictions are placed on the order in which the subtasks are executed. No machine can operate on more than one task at a time, and no task can execute on more than one machine at

the same time. If t_{ij} is the time required by subtask τ_{ij} on machine j, we can obtain the following quantities:

$$\begin{aligned} T_j &= \sum_{i=1}^{n} t_{ij} = \text{ total time on machine } j, \forall 1 \le j \le \eta, \\ L_i &= \sum_{j=1}^{\eta} t_{ij} = \text{ total time for task } i, \forall 1 \le i \le n. \end{aligned}$$

The optimal finish time for all operations is $\alpha = \max_{i,j}\{T_j, L_i\}$, which can always be achieved according to the scheduling algorithm suggested by Gonozalez and Sahni [77]. The scheduling algorithm has a time complexity of $O(\beta^2)$, where β is the number of non-zero subtasks.

Regard all VM-tasks forwarding cells from I_i to O_j as one VM-task (M, C_{ij}), where $C_{ij} \stackrel{def}{=} \sum_{k=1}^{K_{ij}} C_{ijk}$; and regard each output O_j $(j = 1, 2, \ldots, N)$ as a POSS machine. For each given I $(I = 1, 2, \ldots, N)$, regard VM-task subset $\{(M, C_{ij}) | i == I\}$ as a POSS task that runs $C_{I1}, C_{I2}, \ldots, C_{IN}$ time units on POSS machine $O_1, O_2, \ldots, O_N$ respectively. According to the POSS algorithm proposed by Gonzalez and Sahni [77], any feasible VM-task set $\{(M, C_{ij})\}$ can always finish within $\alpha = M$ time units, i.e., any feasible VM-task set $\{(M, C_{ij})\}$ is schedulable; and the scheduling complexity is $O(N^4)$ since $\beta \le N^2$.

2) Necessity: According to the definition given in Constraint 1 and 2, any infeasible VM-task set either exceeds the capacity of an input, or an output, hence is not schedulable.

■

Although Gonzalez and Sahni's POSS algorithm is polynomial and optimal (in the sense it schedules any feasible VM-task set), its implementation is very complex. In the following, we propose a sub-optimal but simpler scheduling algorithm, which has straight-forward graphical meaning.

As in the proof of Theorem 3.2.1, we first regard all VM-tasks forwarding cells from I_i to O_j as one VM-task (M, C_{ij}), where $C_{ij} \stackrel{def}{=} \sum_{k=1}^{K_{ij}} C_{ijk}$. We can graphically represent the VM-task set $\{(M, C_{ij})\}$ $(i, j = 1, 2, \ldots, N)$ as a *demand matrix* (see Fig. 3.2):

Definition 3.2.2 (Demand matrix) *A demand matrix* $\mathbf{D} = \{d_{jg}\}$ *is a* $N \times M$ *matrix, with each element* $d_{jg} \in \{0, 1, 2, \ldots, N\}$. *In the* j*th* $(j = 1, 2, \ldots, N)$ *row,* C_{ij} *elements are colored* i $(i = 1, 2, \ldots, N)$ *respectively; the remaining elements are colored* 0, *meaning* empty slots*; and the elements in the row are arbitrarily ordered.*

In a demand matrix, each non-zero element in the jth row indicates the input from which output O_j shall fetch a cell during a M cell-time clock-period.

Naturally, each demand matrix has the following property:

Property 3.2.3 (Feasible demand matrix) *Suppose the demand matrix* $\{d_{jg}\}_{N \times M}$ *represents a VM-*

task set $\{(M, C_{ij})\}$. *Then* $\{(M, C_{ij})\}$ *is feasible if and only if for each non-zero color* $i \in \{1, 2, \ldots, N\}$, *the demand matrix has no more than* M *elements colored in* i. *Such a demand matrix is called a* feasible demand matrix.

In addition, a demand matrix can represent a schedule.

Definition 3.2.4 (Schedule (matrix)) *We regard a demand matrix* $\mathbf{D} = \{d_{jg}\}_{N \times M}$ *as a schedule if each element* d_{jg} $(d_{jg} \neq 0)$ *implies that output* O_j *grants input* $I_{d_{jg}}$ *at the* g*th cell-time of each* M *cell-time clock-period, and no two elements in each column of* $\mathbf{D}$ *have the same non-zero color. We shall also call such demand matrix a* schedule matrix.

The jth $(j = 1 \sim N)$ row of a schedule matrix represents schedule S_j^{out}. Since a schedule matrix one-to-one mapps to a valid schedule, "schedule matrix" and "schedule" become interchangeable terms.

With the help of the schedule matrix, configuration-time scheduling now has graphical meaning: given a feasible demand matrix $\mathbf{D}$, configuration-time scheduling permutates the elements in each row of $\mathbf{D}$ to produce a schedule (a matrix where no two elements in each column have the same non-zero color). Fig. 3.2 illustrates the relationship between demand matrix, scheduling algorithm, and schedule matrix.

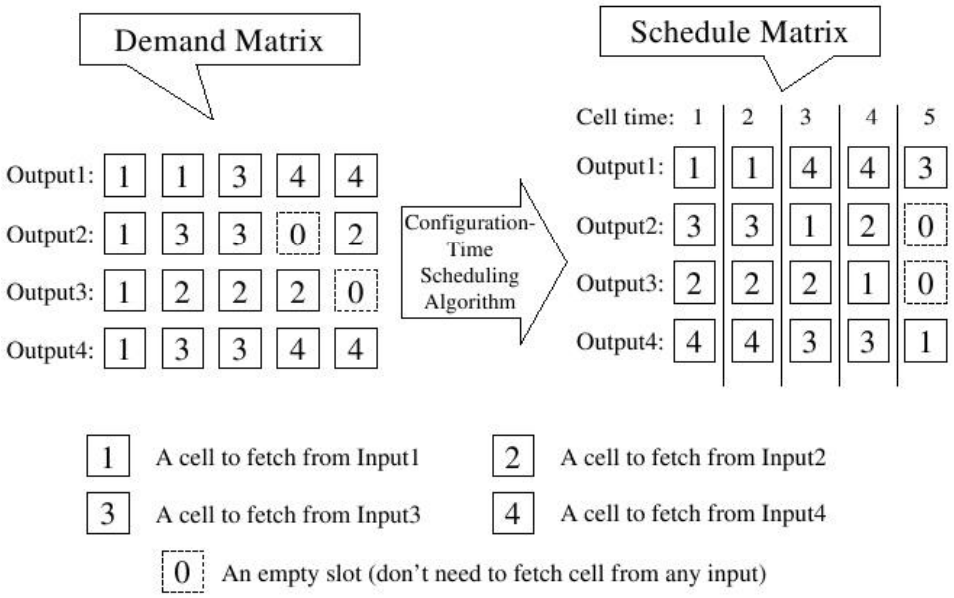


Figure 3.2: An example illustrates the relationship between Demand Matrix, Configuration-Time Scheduling Algorithm, and Schedule Matrix, where number of ports $N = 4$, and a clock-period is $M = 5$ cell-time.

With the help of the above graphical tools, we can devise many simpler sub-optimal scheduling algorithms. In Fig. 3.3, we propose the *least slack* (LS) algorithm. The term "*slack*" means the following: if a row of a demand matrix has κ elements colored c, then color c has a slack of $(M - \kappa)$ in this row.

1. **LeastSlack**(**D**/* the $N \times M$ demand matrix, passed by copy */):
2. Initiate schedule matrix **S** as an $N \times M$ empty matrix.
3. **while D** has non-zero colored element **begin**
4. Of all rows of **D**, pick the non-zero color c that has least slack (break ties arbitrarily).Denote the corresponding row index as j.
5. Move the elements of color c in the jth row of **D** to the earliest (i.e., empty slots with the smallest column indices) and conflict-free empty slots in the jth row of **S**.
break the **while** loop if cannot find any conflict-free empty slot.
6. **end**.
7. **if** all non-zero colored elements of **D** are removed, **return S**;
8. **else return** cannot find schedule.

Figure 3.3: Least Slack (LS) Scheduling. The term "conflict-free" means no two non-zero colored elements in each column of a matrix have the same color.

For LS-scheduling algorithm, let tuple (r, c) correspond to the slack of color c in the rth row of demand matrix. During initialization, we shall create and sort these N^2 tuples into a list $\mathcal{L}$ with ascending slack, which takes $O(N^2 \log N + NM)$ time. Then Step 3 only takes $O(1)$ time: just to check whether $\mathcal{L}$ is empty; and Step 4 only takes $O(1)$ time: just remove the head of $\mathcal{L}$. Step 5 takes $O(M)$ time, if we maintain an $N \times M$ boolean array F for **S** with F_{cg} indicating whether the gth column of **S** already has an element colored c. The **while** loop from Step 3 to Step 6 loops at the most N^2 times. Therefore, the time complexity of LS-scheduling is $O(N^2 \log N + NM + N^2 M) = O(N^2 M)$.

3.2.5 E2E Delay Guarantee

In this section, we analyze the E2E delay guarantee provided by our proposed real-time switch for industrial real-time applications. In these applications, the dominate traffic is periodic, such as sensing, actuating, and video monitoring. Aperiodic traffic can be served by periodic VM-tasks [65]. As a result, we shall assume all traffic is periodic in the following analysis.

We assume that all the switches in the industrial network comply with the proposed real-time switch scheme. We also assume that all switches adopt the same clock-period of $\mathcal{P} \equiv 1$ (ms) and have the same per port capacity. Assume a uniform cell size of 500 bits[1]. If the per port capacity is 1Gbps, 10Gbps, or 100Gbps, then a clock-period of 1ms corresponds to an M of 2000, 20000, and 200000 cell-time respectively.

Suppose that a real-time flow f needs to send, at the least, a message of E cells every T cell-time, denoted as $f = (T, E)$. Note E, T may be real numbers instead of integers. Then we over provision f with VM-task $\tau_f = (M, C)$, where

$$C = \left\lceil \frac{E}{\lfloor T/M \rfloor} \right\rceil . \tag{3.4}$$

That is, each message of f is forwarded as $R \stackrel{def}{=} \lfloor T/M \rfloor$ packets, and each packet consists of C cells.

[1]Real-world switches usually use cell size of 512 bits. We use cell size of 500 bits for narrative simplicity.

Note, since M cell-time equals 1ms, for most industrial real-time applications, $T > M$.

Suppose f traverses H hops of our proposed real-time switches, each schedules a VM-task of (M, C) to forward the packets of f.

To derive the E2E delay, we start from the first hop. Since the first hop forwards exactly C cells for flow f in any consecutive M cell-time, whenever a new message of f arrives, the first packet of the message takes at the most $M + 1$ cell-time to be forwarded, the additional 1 is because the packet may arrive during the middle of a cell-time. After that, the switch forwards a next packet every additional M cell-time, until all R packets are forwarded. Same thing happens in the following switches. Therefore, the worst case E2E delay D (ms) for the message is

$$\begin{aligned} D &= \sum_{h=1}^{H}(M+1)\delta + (R-1)M\delta \\ &= (H+R-1)\mathcal{P} + H\delta, \end{aligned} \tag{3.5}$$

where δ (ms) is one cell-time in the unit of millisecond.

The first item of Equation (3.5) is the worst case E2E delay for the first packet. After the first packet arrives at the receiver end, every additional M cell-time, a subsequent packet arrives, until all R packets arrive.

Note the above analysis can be easily extended to cases where the proposed real-time switches have different per port capacities, which are not discussed in this paper due to page limits.

3.3 Evaluation

3.3.1 Efficiency of M Cell-Time Clock-Period

A natural question on the proposed real-time switch is: how efficient is it to enforce a unanimous M cell-time clock-period?

We evaluate this in the context of industrial real-time control/automation traffic.

There are two types of real-time traffic in industrial real-time control/automation: real-time sensing/actuating traffic and real-time video traffic.

Real-time sensing/actuating traffic involves low data-throughput. A typical sensing/actuating flow generates a $1 \sim 5$kbit message every 10(ms). The maximal allowed E2E delay is usually 50ms [78, 79].

Real-time video traffic involves high data-throughput. A typical video flow generates one message (a.k.a. "frame") every 30ms, and the message size is in the worst case $120 \sim 240$kbits. And usually the E2E delay

for each video frame is also 50ms [78, 79].

As in Section 3.2.5, we assume a fixed cell size of 500bits/cell, and we always pick M so that M cell-time equals 1ms.

In the following, we run 1000 trials for each type of switch settings: with per port capacity of 1Gbps, 10Gbps, and 100Gbps; and number of input ports (which is also the number of output ports) of 8, 16, and 32.

In each trial, we randomly add sensing/actuating or video flows to a switch (without exceeding port capacities); and the messages of each flow f are over-provisioned with VM-task (M, C) as described in Equation (3.4) of Section 3.2.5. For each flow set, we calculate its switch utilization demand, and check whether the flow set is schedulable using the M cell-time clock-period. Note that the switch utilization demand is calculated using each flow's original message period and message size, not the over-provisioned VM-task (M, C); and switch utilization equals the average utilization of all inputs of the switch (assume all inputs has the same capacity). Fig. 3.4 plots the schedulability ratio (i.e. probability) for given switch utilization demand.

We find that our real-time switch achieves good schedulability and switch utilization. When the switch utilization demand is below 70%, a flow set is empirically always schedulable in all settings. Particularly, for high-speed switches with per port capacity of 10Gbps and 100Gbps, the switch utilization can reach nearly 85% and 90% for all settings to provide a 100% schedulable ratio (empirically).

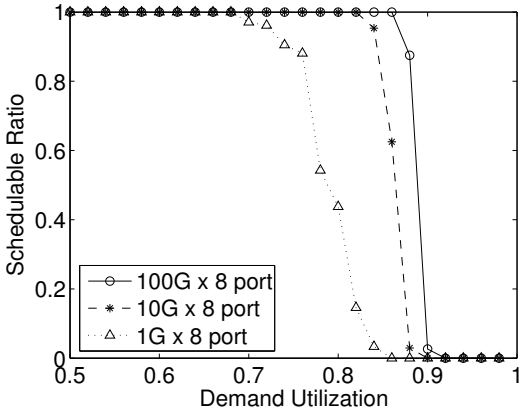
We also find that the M cell-time clock-period schedulability ratio improves as per-port capacity increases. Take Fig. 3.4 (a) for example: a switch utilization demand of 86% corresponds to a schedulability ratio of 0, 96%, and 100% when the per port capacity is 1Gbps, 10Gbps, and 100Gbps respectively.

On the other hand, the schedulability ratio deteriorates as the number of ports increases. For example, the 1Gbps curves of Fig. 3.4 (a), (b), and (c) shows that when the switch utilization demand is 80%, the schedulability is 43%, 22%, and 0 for 8 port, 16 port, and 32 port switches respectively. This is intuitive because more ports means more contention.

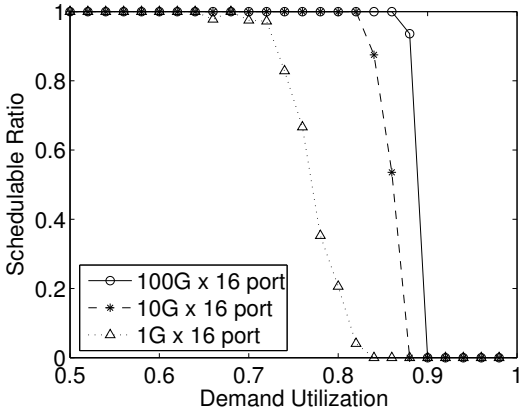
3.3.2 E2E Delay

The same simulation study described in Section 3.3.1 also provides E2E delay upper bound statistics. We compare them with those of *i*SLIP.

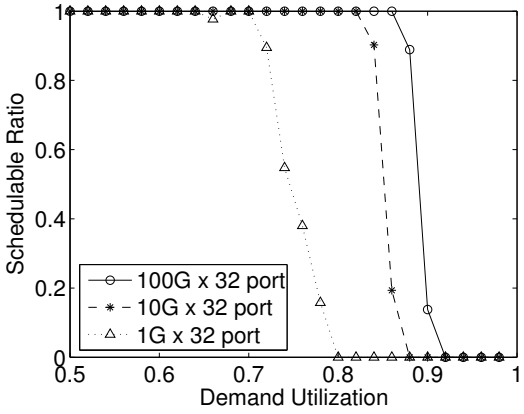
We assume the maximal hop count is 15. The E2E delay upper bound of our proposed real-time switch is given in Equation (3.5). A tight *i*SLIP E2E delay bound, however, is still an open problem. To make the comparison optimistic on the *i*SLIP side, we use the *i*SLIP *single hop* delay bound given in Equation (3.1)



(a)



(b)



(c)

Figure 3.4: Schedulability Ratio for Given Switch Utilization Demand using the Proposed Real-Time Switch and M Cell-Time Clock-Period

as its E2E delay bound.

The result statistics are shown in Fig. 3.5.

We see that using our proposed real-time switch, all E2E delays are within 50ms, which meets the demand of most industrial real-time traffic. Using *i*SLIP switches, however, most of the time even the single hop delay bound *may* exceed 100ms, 150ms, or even 200ms. Therefore, our proposed real-time switch provides better E2E delay guarantees.

3.3.3 Efficiency of LS Algorithm

Lastly, we evaluate the efficiency of LS algorithm described in Fig. 3.3.

We know that Gonzalez and Sahni's POSS algorithm is optimal in the sense that it can schedule any feasible demand matrix. LS is a simpler, but sub-optimal algorithm. For any feasible demand matrix, POSS provides a schedulability ratio of 100%. We compare this with LS's schedulability ratio. We still try three different numbers of ports: 8, 16, and 32. For each number of ports, we try three different per port capacity: 1Gbps, 10Gbps, and 100Gbps. For each setting, we randomly generate 1000 feasible demand matrices, and check whether they are schedulable using LS algorithm. The results are plotted in Fig. 3.6.

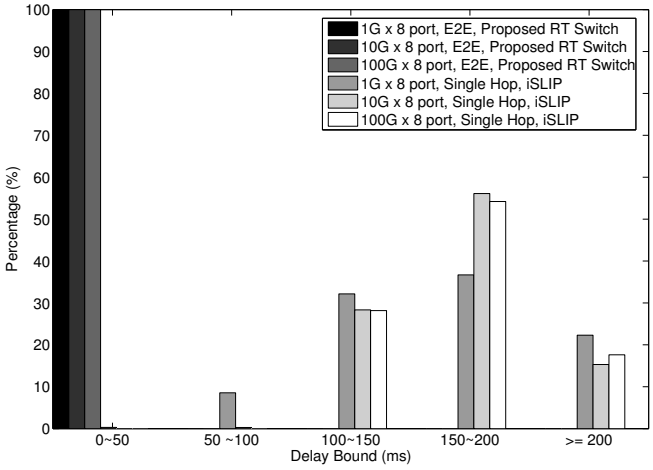
We find that LS schedulability is sensitive to number of ports. As shown in Fig. 3.6 (a), (b), and (c), as number of ports increases from 8, to 16, and to 32, the LS-algorithm can schedule more than half, about half, and less than half of the randomly generated feasible matrices. This is intuitive because more number of ports means a demand matrix has more colors to conflict with each other in each column.

We also see that LS schedulability is not sensitive to per port capacity: in all of Fig. 3.6 (a), (b), and (c), different per port capacity of 1Gbps, 10Gbps, and 100Gbps result in similar curves. This is probably because the number of colors that can conflict is fixed, given the number of ports is fixed.

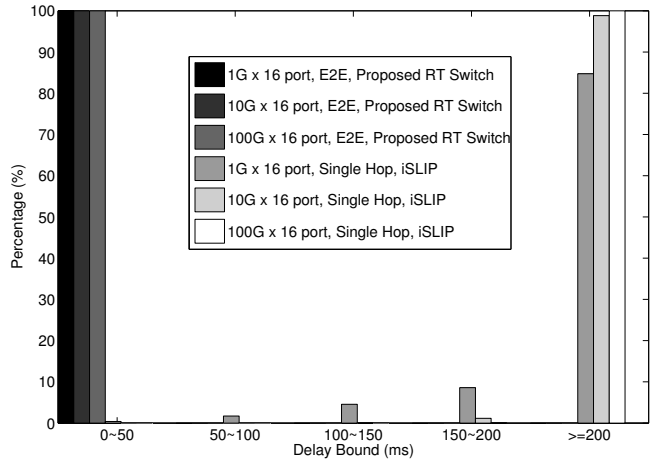
3.4 Related Work

Network infrastructure for hard real-time communication has typically been restricted to prioritization in routers. The number of priority levels, however, is about 4 to 8 in conventional Internet routers, and this is insufficient for hard real-time guarantees. Additionally, many router designs for real-time systems have required significant changes when compared to commercially-available routers for Internet traffic. The desire to use existing solutions, or solutions with minimal hardware changes, has been a dominant interest for industrial networks from the viewpoint of purchasing and maintenance costs.

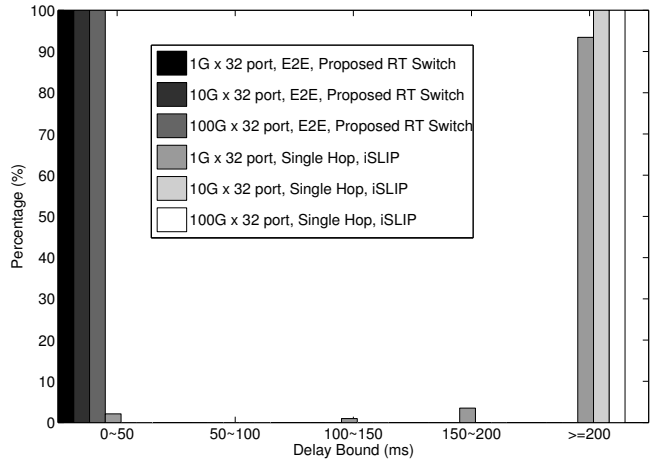
Prioritized bus and ring networks have been used in small real-time systems [80–82] but they are not



(a)

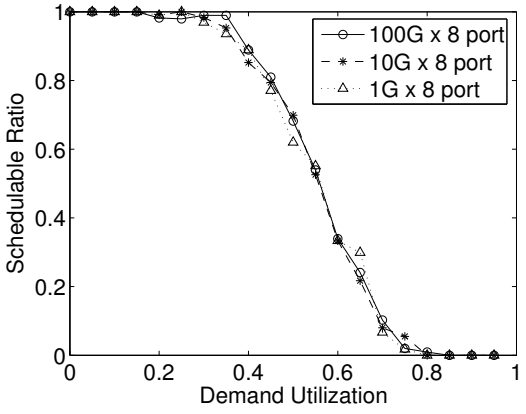


(b)

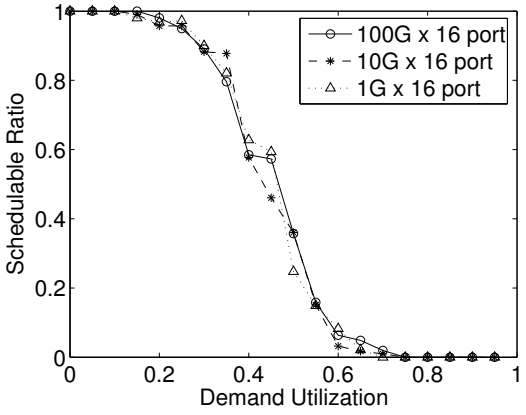


(c)

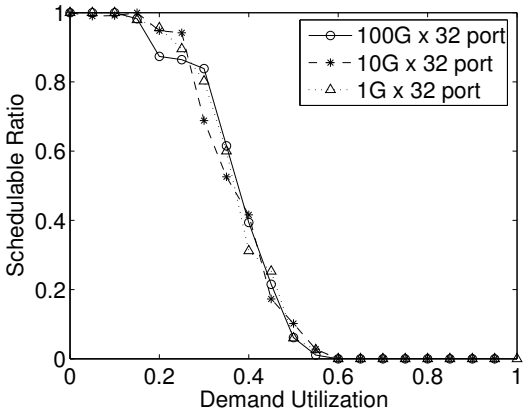
Figure 3.5: E2E Delay Comparison



(a)



(b)



(c)

Figure 3.6: LS Schedulability Ratio for Given Demand Matrix Utilization

designed for high-speed network backbones, such as those of WANs. Rexford, Hall and Shin [83] propose a router for real-time communication but it was designed to support deadline-based scheduling, which imposes significant hardware changes. Additionally, their router is not designed for high-speed network backbones either. Similarly, Venkatramani and Chiueh proposed a real-time switch for Ethernets [84], which is neither designed for high-speed network backbones.

While there has been some effort, such as by Rexford, Hall and Shin, to design new routers for real-time systems, considerable effort has been devoted to analyzing the performance of high-speed switches and routers and obtaining delay bounds [85, 86]. The scheduling of crossbar switches reduces to a matching on a graph, and fast algorithms for obtaining a matching have also been studied [87]. These results use stochastic traffic patterns and provide asymptotic performance bounds that are not sufficient for industrial systems that require greater predictability.

Some related work concerns the use of COTS routers for real-time systems using approximate bounds and designing networks of switches to meet end-to-end deadlines [69]. The work presented in this article complements such work; better router architectures result in reduced message delays, which in turn reduces the cost of networks that can guarantee end-to-end requirements.

There are also efforts on emulating output queueing using input queueing or combined input-output queueing [88, 89]. However, to achieve the same hardware utilization efficiency as that of conventional input-queueing/VOQ crossbar switches is still an open problem.

The work presented in this article provides a mechanism for guaranteeing a task a certain amount of communication slots in a fixed time interval. The router design we have articulated is a building block for obtaining end-to-end delay bounds, and for enabling hierarchical scheduling policies and associated analysis [70–72, 74].

The content of this chapter is published in [90].

3.5 Conclusion

The convergence of computer and physical world is the theme for next generation networking research. This trend calls for real-time industrial network infrastructure, which needs high-speed real-time WAN to serve as its backbone. However, nowadays commercially available high-speed WAN switches (routers) are designed for best-effort Internet traffic. A real-time switch design for the aforementioned networks is missing.

In this article, we propose a real-time switch design based on the most widely adopted crossbar switch architecture. The proposed switch can be implemented by making minimal modifications, or even simplifi-

cations, to the well-known *i*SLIP crossbar switch scheme. This benefits switch manufacturers since *i*SLIP is already widely implemented in commercial products, and the minor modifications can be easily incorportated into the manufacturing process.

Our real-time switch serves periodic and aperiodic traffic with real-time virtual machine tasks, which simplifies analysis, provides isolation, and facilitates future hierarchical scheduling and flow aggregation. Taking advantage of the fact that most industrial real-time network flows rarely change, the switch only needs to be configured to a real-time schedule at startup-time (aperiodic flows, which may change more frequently, are encapsulated by their real-time virtual machine tasks), and a polynomial time algorithm is found to schedule any feasible flow set. During runtime, our real-time switch incurs only $O(1)$ computation, which fits the need of high-speed networking.

Simulation results show that, for typical industrial real-time network traffic, our switch can achieve high utilization and guarantee small end-to-end delays.

We believe that it is essential to capture the true workload characteristics of applications, such as the predictability of network traffic in industrial control applications, to design efficient infrastructure for these applications. Further, changes in workload, which are infrequent and involve planned outages, can be accommodated via simple reconfiguration. As future work, we will extend our switch design to support run-time adaptation, hierarchical scheduling, and flow aggregation. We are also interested in better analyses for end-to-end delay bounds, and in resource optimization issues.

Chapter 4

REAL-TIME ACOUSTIC EVENT LOCALIZATION IN WIRELESS SENSOR NETWORKS

In addition to the RT-WLAN plus real-time wired backbone model, *Wireless Sensor Network* (WSN) is another important infrastructure for CPS applications. WSNs can be widely used for CPS applications such as surveillance, structure and environment monitoring, and assisted living. One basic service in WSNs is localization of acoustic events.

In this chapter, we are interested in electing the closest sensor to an acoustic event. Such election serves the purpose of *proximity-based localization*, i.e., giving the location of the elected sensor as the approximate location of the acoustic event. For many applications where wireless sensors are densely deployed, proximity-based localization provides sufficient resolution. Electing the closest sensor to the acoustic event may also serve the purpose of runtime *leader election* for collaborative sensing. For example, many fine-grain localization (triangulation) algorithms [91, 92] require collaboration of sensors in the vicinity of the acoustic event, and a unique leader must be elected to start such collaboration. A protocol that quickly elects the closest sensor to the acoustic event serves this purpose.

An acoustic event localization (simplified as "acoustic localization" in the following) protocol in a dense WSN shall desirably come with following properties.

- **Timeliness:** Many applications demand locating acoustic events in short and bounded time. Particularly, a hard real-time application, such as gunshot localization (see [93]), requires a short and constant ($O(1)$) localization time bound.
- **Lightweightness:** Low-end sensors have constrained computation, storage and communication capabilities. Therefore, it is desirable for acoustic localization algorithms to be simple and efficient.
- **Reasonable Accuracy:** The accuracy of localization must be acceptable, but not excessive. For many applications, proximity-based localization is sufficient.
- **Robustness:** The scheme shall work with unreliable wireless medium and irregular acoustic signals: sound intensity may be direction-dependent and sound propagation may be affected by multi-path effects.

- **Energy-Efficiency:** An energy-efficient localization algorithm should have minimal message exchanges in presence of acoustic events and conserve energy when there is none.

To achieve the above goals, we propose *Lightning Protocol*, a hard real-time, fast, and lightweight wireless sensor election protocol, for locating impulsive acoustic events using low-end wireless sensors. Lightning Protocol exploits the fact that electromagnetic wave propagates much faster than acoustic wave to localize acoustic events in a proximity-based fashion. Like many other acoustic localization protocols, it utilizes acoustic *Time-Of-Arrival* (TOA) information at sensors. However, *Lightning Protocol does not send data packets; nor does it require clock synchronization.* Instead, it allows overlapping wireless broadcasts, which greatly simplifies the implementation. Lightning Protocol exhibits the afore-mentioned desirable properties and fits indoor or open area deployment.

Lightning Protocol is implemented on U.C. Berkeley Motes [94, 95] and compared with an ideal *Data Packet* (DP)-based scheme. Using Lightning protocol, the entire machine code only occupies 5330 bytes in ROM and 187 bytes in RAM. In Section 4.5, experimental results show that the localization time cost of Lightning Protocol is less than 3.6ms. Among 81.4% of the localization trials, there is only *one* wireless broadcast involved (with the rest being suppressed). The accuracy is comparable to or better than that of an ideal collision-free DP scheme.

The rest of the chapter is organized as follows. In Section 4.1, we motivate the proposed solution by closely examining the acoustic characteristics in our application scenarios. The basic Lightning Protocol and its improved versions (energy-efficient, random layout) are described in Section 4.2, 4.3 and 4.4 respectively. Experimental results are presented in Section 4.5, followed by discussions in Section 4.6. Related works are discussed in Section 4.7. Section 4.8 concludes the chapter.

4.1 Assumptions and Design Considerations

We assume following properties on the acoustic signals:

P1 Impulsive Sound: The acoustic signal shall come with clearly detectable onset and thus their TOAs can be measured. We call such acoustic signals *impulsive sounds*, or *beeps*.

P2 Bounded Directionality: The intensity of a beep may be directional, that is, "loudest" does not necessarily mean "closest". Meanwhile, the directionality is bounded: the closest sensor shall be able to detect the TOA of the sound (a quantitative definition is given in Section 4.2.3).

P3 Moderate Multipath: An acoustic signal may travel along indirect paths to a sensor; but the closest sensor shall have *Line-Of-Sight* (LOS) to the sound source.

Properties P2 and P3 naturally promote the use of *densely* placed sensors, where LOS is usually available between the sound source and its closest sensor, and hence the acoustic wave would arrive at the closest sensor before echoes. Properties P1 and P2 make TOA a better parameter than acoustic signal intensity for localization: for impulsive sound sources, as long as LOS to the closest sensor is available, TOA of the closest sensor is earlier than TOAs at other sensors, independent of the directionality and loudness of the sound.

Other assumptions are as follows:

A1 Uniform Acoustic Medium: The sound propagation medium (e.g., the air) is stable and ensures uniform acoustic propagation speed, denoted by v.

A2 Wireless Range: Wireless transmission range is at least twice as large as the acoustic sensing range. Therefore, all sensors that overhear the same sound can directly communicate with each other[1].

A3 RF Device: Each sensor is equipped with a half-duplex radio transceiver with configurable *Radio-Frequency* (RF) channel that can be changed during runtime.

A4 Deployment Environment: We assume Lightning Protocol runs in open space or lightly obstructed office environment, without severe acoustic multipath effect. Although significant acoustic multipath effects can be handled with extra time cost (see Section 4.6.3), they are not the focus of this paper.

4.2 Basic Lightning Protocol

4.2.1 Intuition

As mentioned in Section 4.1, TOA measurements based on onset detection are more robust to directionality of acoustic signals. A straightforward election algorithm is to compare the TOAs at each sensor. The one with the earliest TOA reading is considered the closest to the sound source. However, this approach suffers two problems, (*i*) packet communication overhead that grows with increasing sensor density; and (*ii*) requirement of clock synchronization as each sensor needs to time stamp the TOA based on its local clock.

Our proposed Lightning Protocol is self-synchronized and only involves $O(1)$ RF broadcasts to elect the closest sensor. The intuition comes from lightning phenomenon in nature. When a lightning bolt strikes,

[1]This can be achieved by properly adjust the threshold for beep detection.

people *see* the lightning much earlier than they *hear* the accompanying thunder, because electromagnetic waves travel much faster than sound waves. To apply this property to our election problem, we notice that an acoustic signal arrives at the closest sensor (denoted as S_1) first (note the dense deployment of sensors empirically guarantees the availability of LOS). If S_1 can *immediately* transmit an RF signal to notify all other sensors, they can decide that they are farther away from the sound source even before the acoustic wave reaches them. Nonetheless, there is still one difficulty: wireless transmissions are usually subject to collisions. In particular, if there are multiple sensors at similar distances to the sound source, broadcasting data packets suffer from collisions that prevent an immediate notification. To solve this problem, we propose using raw RF burst to signal the arrival of sound wave. RF bursts can overlap, which obviates the random backoff based wireless MAC, and makes the protocol immune to wireless broadcast collisions. RF bursts can be detected by measuring the received radio energy [96].

Based on the above ideas, we propose Basic Lightning Protocol in the next subsection.

4.2.2 Protocol Details

For the time being, we assume sensors are placed at regular grids on the plane (Fig. 4.1). Without loss of generality, we assume square grid as shown in Fig. 4.1. Extension of Lightning Protocol to *random layouts* is presented in Section 4.4. In a regular grid layout, each sensor is assigned a color i, as shown in Fig. 4.1 (for square, $i \in \{1, 2, 3, 4\}$). The color assignment guarantees that for any point in the plane, the enclosing sensors are of distinct colors. As is to be explained later, such a color assignment guarantees the uniqueness of the elected sensor.

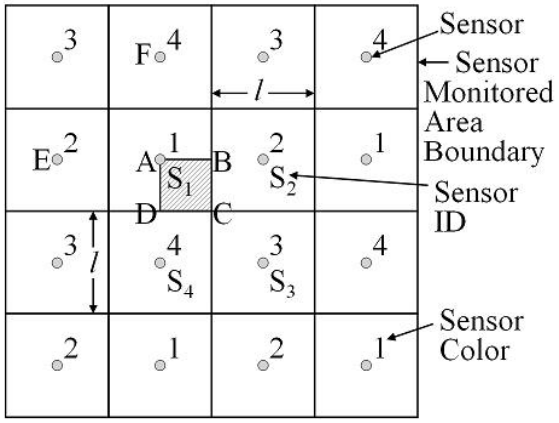


Figure 4.1: A Square Sensor Grid Layout

Let T_b be the minimum duration that an RF burst must sustain in order to be robustly detected. The

Basic Lightning Protocol goes as follows:

1. All sensors are initially in RF *listening mode*.

2. While in the listening mode, if a beep is *recognized*[2], a sensor with color i switches immediately to RF *bursting mode* and broadcasts *without back-off* an RF burst of $i \cdot T_{burst}$ duration (T_{burst} is an implementation-specific constant).

   Immediately after the burst, the sensor switches back to RF listening mode and samples the wireless medium for T_b. If no other RF burst is recognized, it wins the election and enters the *elected mode*. Otherwise, it loses the election and enters the *suppressed mode*. In both cases, a sensor sets up a timer of length T_{reset}^{basic}.

3. At any time during RF listening mode, if an RF burst is recognized, the sensor fails the election and enters the suppressed mode. Meanwhile, the sensor sets up a timer of length T_{reset}^{basic}.

4. After the election is completed (when timer T_{reset}^{basic} expires), all sensors return to the listening mode.

T_{reset}^{basic} is a preset constant for all sensors to reset their state meanwhile ensure consistency. The setting of T_{reset}^{basic} is explained later.

As mentioned earlier in Section 4.1, we assume sensors can switch between multiple RF channels during runtime. This allows us to use a separate RF channel for burst. When a sensor is not elected, it only listens and bursts at the RF burst channel. Once elected, a sensor may switch to data communication channel to conduct data exchanges. Fig. 4.2 summarizes the state transitions of the Basic Lightning Protocol.

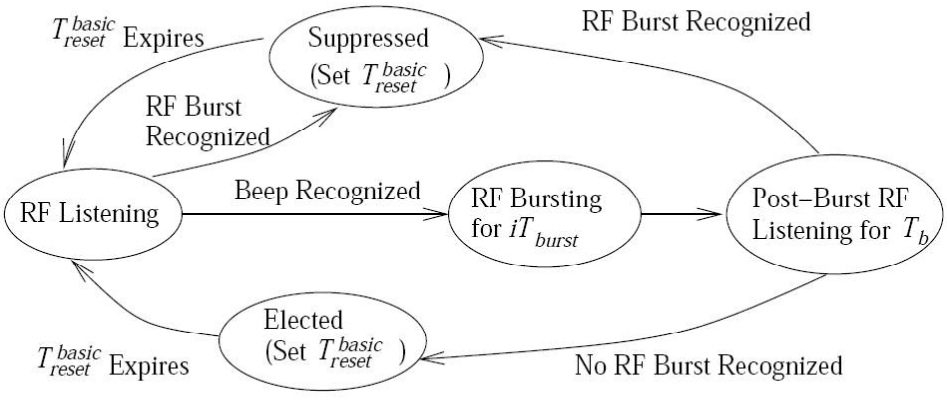


Figure 4.2: State Transition Diagram of Basic Lightning Protocol for a Sensor with Color i

[2] "*Recognize*" refers to the time instance when the arriving beep is detected by the sensor's signal recognition module and reported to the sensor's election module.

4.2.3 Properties of Basic Lightning Protocol

In this section, we prove that Basic Lightning Protocol elects a unique sensor within $O(1)$ time delay, and the elected sensor is among the closest sensors enclosing the sound source.

For now, we assume there is only one acoustic event (beep). The case of multiple sound events is discussed in Section 4.6.1. *Without loss of generality, we consider a square sensor grid, and the sound source location* $p = (x, y) \in \square ABCD$ *(see Fig. 4.1,* $\square ABCD$ *refers to the shaded square area).* The distances between the sound source and the four adjacent sensors S_1, S_2, S_3 and S_4 (colored 1,2,3 and 4 respectively) are $d_1(p)$, $d_2(p)$, $d_3(p)$ and $d_4(p)$ respectively. $d_{other}(p)$ is the distance between the sound source location p and the closest sensor other than $S_1 \sim S_4$. We assume the sensor density is sufficiently high so that at any position $p \in \square ABCD$, a beep is recognizable to all sensors $S_1 \sim S_4$ regardless of its directionality. As RF waves travel a lot faster than acoustic signals, we ignore the propagation delay of RF burst. Let t_{recg} be units of time for a sensor to *recognize* a beep, $t_{recg} \in [0, \Delta_{recg}]$, i.e. Δ_{recg} is the maximum time cost to recognize a beep. The notations used in the analysis are summarized in Table 4.1.

Table 4.1: Notation used in the analysis

t_{recg}	a random variable denoting the time cost to recognize a beep
$0, \Delta_{recg}$	the minimum and maximum possible time to recognize a beep (Δ_{recg} can be 0.12ms)
T_{burst}	the RF burst duration for a sensor colored 1 (can be 0.8ms)
iT_{burst}	the RF burst duration for a sensor colored i
T_b	minimum duration that an RF burst must sustain in order to be recognized (can be 0.4ms)
$T^{\star}_{elect}$	election delay, $\star$ corresponds to *light* for Lightning Protocol, or *data* for DP protocol.
$T^{\star}_{reset}$	time to reset to initial mode from suppressed or elected mode. $\star$ corresponds to a specific version of Lightning Protocol
$T^{\star}_{bound}$	election delay bound, the time bound from the beep takes place to all sensors enter elected or suppressed mode. $\star$ corresponds to a specific version of Lightning Protocol
$T^{\star}_{bound}+T^{\star}_{reset}$	turn-around time
R^{max}_{beep}	the maximum audible radius of a beep
p	(x, y) location of the sound source, $p \in \square ABCD$, see Fig. 4.1
l	grid edge length, see Fig. 4.1
$d_1(p) \sim d_4(p)$	distance between the sound source ($p \in \square ABCD$) to the four adjacent sensors ($S_1 \sim S_4$) in Fig. 4.1.
$d_{other}(p)$	distance between the sound source and the closest sensor other than the four adjacent sensors
$d(p)$	distance between the sound source and the elected sensor
v	sound speed (1ft/ms)

Before delving into the derivation, we first present the key results. Theorem 4.2.1 states that Basic

Lightning Protocol elects a single sensor and the localization error is bounded; Corollary 4.2.3 gives the upper bound of *election delay* (see Definition 4.2.2).

Theorem 4.2.1 (Winner Uniqueness and Error Bound) *If $T_{burst} \geq 2T_b$ and square grid edge length $l > \frac{2}{2-\sqrt{2}}(\Delta_{recg} + T_b)v$, Basic Lightning Protocol elects* exactly one *sensor, which is among $S_1 \sim S_4$. Furthermore, the distance $d(p)$ between the sound source location p and the elected sensor satisfies $d(p) \leq d_1(p) + (\Delta_{recg} + T_b)v$.*

Definition 4.2.2 (Election Delay) *Election Delay refers to the time duration since the beep takes place till every sensor enters either elected or suppressed mode.*

Corollary 4.2.3 (Election Delay Bound) *If $T_{burst} \geq 2T_b$ and $l > \frac{2}{2-\sqrt{2}}(\Delta_{recg} + T_b)v$, Basic Lightning Protocol incurs an election delay no greater than $T^{basic}_{bound} = \frac{\sqrt{2}l}{2v} + 2\Delta_{recg} + 4T_{burst} + 2T_b$.*

To prove the above results, we first present the following lemmas.

Lemma 4.2.4 *If square edge length $l > \frac{2}{2-\sqrt{2}}(\Delta_{recg} + T_b)v$ and $T_{burst} > T_b$, then only S_1, S_2, S_3 and S_4 may burst, all other sensors are suppressed.*

Proof: Recall that $d_{other}(p)$ is the distance between p and the closest sensor other than $S_1 \sim S_4$. $\inf_{\forall p \in \square ABCD}\{d_{other}(p)\} = l$, and this value is reached *only* when $p = A$. The sensor is either at point E or F (see Fig. 4.1). On the other hand, $\sup_{\forall p \in \square ABCD}\{d_1(p)\} = \frac{\sqrt{2}}{2}l$, and this value is reached *only* when $p = C$. Hence, $\forall p \in \square ABCD$, $d_{other}(p) - d_1(p) > \frac{2-\sqrt{2}}{2}l$ (note "=" can never be achieved because that requires p be at A and C at the same time). When $l > \frac{2}{2-\sqrt{2}}(\Delta_{recg} + T_b)v$, we have $\forall p \in \square ABCD$, $\frac{d_{other}(p) - d_1(p)}{v} > (\Delta_{recg} + T_b)$. Hence, when the sound wave reaches any of the sensors other than $S_1 \sim S_4$, S_1 either has recognized the beep and burst for at least T_b units of time, or has been suppressed by a burst from S_2, S_3 or S_4. In both cases, any sensor other than $S_1 \sim S_4$ is suppressed. ■

Definition 4.2.5 (Non-Deterministic Area) *When the beep is at a location that is equidistant or nearly equidistant to multiple sensors, due to randomness of beep recognition time cost t_{recg}, the closest sensor may not always burst first and suppress other sensors, i.e. the elected sensor may be not the closest sensor. Such locations form a* Non-Deterministic Area*. However, we shall see the localization error incurred by*

non-deterministic area is small *(Theorem 4.2.1). Specifically, we define non-deterministic area V as:*

$$V \stackrel{def}{=} \Big\{p \Big| p \in \square ABCD \text{ and } \frac{d_1(p)}{v} + \Delta_{recg} + T_b \geq \min\{\frac{d_2(p)}{v}, \frac{d_3(p)}{v}, \frac{d_4(p)}{v}\}\Big\}$$

Note as mentioned in the beginning of this subsection: *without loss of generality*, we only consider the cases that $p \in \square ABCD$. The above definition of V therefore only refers to the non-deterministic area *within* $\square ABCD$. For areas other than $\square ABCD$, the definition of non-deterministic area follows the same pattern, with $d_1(p)$ replaced with distance to the closest sensor, and $d_2(p), d_3(p), d_4(p)$ replaced with distances to the other three enclosing sensors.

Lemma 4.2.6 *When $l > \frac{2}{2-\sqrt{2}}(\Delta_{recg} + T_b)v$ and $T_{burst} > T_b$, if $p \in \square ABCD$ and $p \notin V$, then only S_1 would burst and thus win the election.*

Proof: By Lemma 4.2.4, the only sensors that may compete with S_1 are S_2, S_3 and S_4. $p \in \square ABCD$ and $p \notin V$ means $\frac{d_1(p)}{v} + \Delta_{recg} + T_b < \min\Big\{\frac{d_2(p)}{v}, \frac{d_3(p)}{v}, \frac{d_4(p)}{v}\Big\}$. Therefore, by time $d_1(p)/v + \Delta_{recg}$, S_1 have recognized the beep and started bursting, meanwhile the beep has not yet reached S_2, S_3 and S_4. By time $\frac{d_1(p)}{v} + \Delta_{recg} + T_b$, S_1 have already burst for T_b duration, while the beep has not yet reached any of S_2, S_3 and S_4. Therefore, $S_2 \sim S_4$ are suppressed by S_1 (note since $S_2 \sim S_4$ are in listening mode, a burst duration of T_b from S_1 is long enough to suppress $S_2 \sim S_4$). ■

Likewise, we can prove that if $p \in V$, S_2, S_3 or S_4 may burst before the closest sensor S_1, or even suppress S_1. In Fig. 4.3, we plot non-deterministic area V for square $\square ABCD$ (edge-length equals 2 ft) under different $\Delta_{recg} + T_b$. To prove Theorem 4.2.1, note that by the color assignment shown in Fig. 4.1, $S_1 \sim S_4$ each has a distinct color. Without loss of generality, suppose S_1, S_2, S_3 and S_4 are of color 1, 2, 3 and 4 respectively. Then we have the following lemma:

Lemma 4.2.7 *If $T_{burst} \geq 2T_b$, then after a beep, if multiple sensors burst, the one with "largest color number" always wins the election.*

Proof: According to Lemma 4.2.4, if multiple sensors burst after a beep, the total number of bursting sensors can only be 2, 3 or 4.

Case 1: *2 sensors burst.* We denote them as S_i and S_j, where $i, j \in \{1, 2, 3, 4\}$ and $i > j$. Suppose S_i bursts during time interval $[t_i^0, t_i^0 + iT_{burst}]$ and S_j bursts during $[t_j^0, t_j^0 + jT_{burst}]$. Then we must have

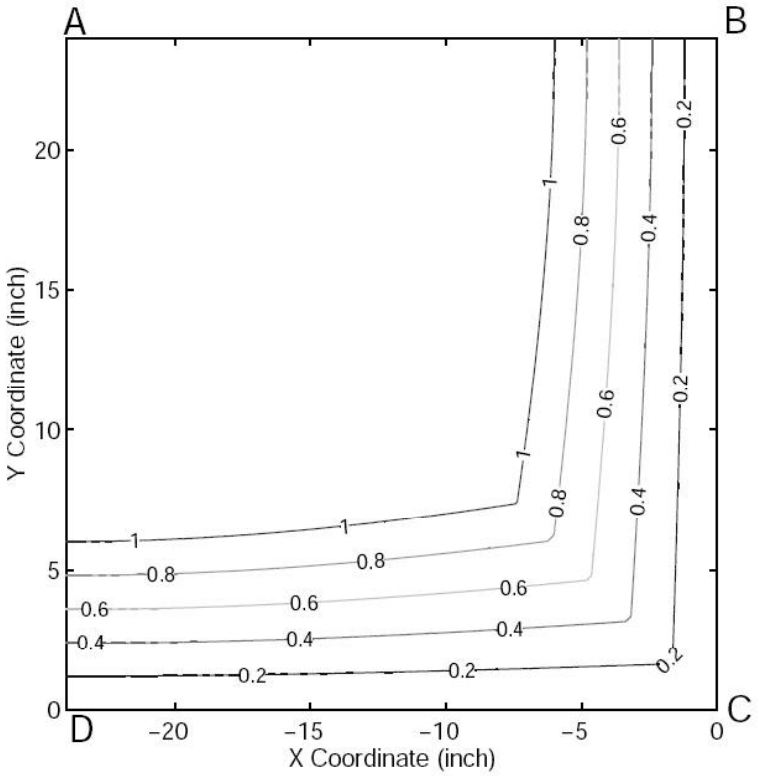


Figure 4.3: Area lower-right to the contour is the non-deterministic area within $\square ABCD$ of Fig. 4.1, where $\Delta_{recg} + T_b = 0.2, 0.4, 0.6, 0.8$ and 1ms respectively. Assume sound propagation speed $v = 1$ft/ms and edge-length of $\square ABCD$ to be 2ft.

$|t_i^0 - t_j^0| < T_b$, otherwise, the later burst would be suppressed. (*i*) If $t_i^0 < t_j^0$, after S_j stops bursting, burst from S_i will last for

$$\begin{aligned} & iT_{burst} - jT_{burst} - (t_j^0 - t_i^0) \\ = \; & (i-j)T_{burst} - (t_j^0 - t_i^0) \\ > \; & T_{burst} - T_b \geq 2T_b - T_b = T_b \end{aligned}$$

The last "$\geq$" is due to $T_{burst} \geq 2T_b$. Hence, S_j has enough time to recognize S_i's burst after its own burst, and to realize that it has lost the election. (*ii*) If $t_i^0 \geq t_j^0$, after S_j stops bursting, burst from S_i will last for $iT_{burst} - (jT_{burst} - (t_i^0 - t_j^0)) = (i-j)T_{burst} + (t_i^0 - t_j^0) \geq T_{burst} \geq 2T_b > T_b$. Hence, S_j has enough time to recognize S_i's burst after its own burst, and realize that it has lost the election. Therefore, according to (*i*)(*ii*), S_i would always win.

Case 2: *3 and 4 sensors burst.* The same reasoning as in Case 1 can be applied here. ■

Now we are in the position to prove Theorem 4.2.1. Theorem 4.2.1 states that a unique sensor is elected

in Basic Lightning Protocol and the localization error is bounded.

Proof: By Lemma 4.2.4, the only possible competing sensors are $S_1 \sim S_4$. By Lemma 4.2.6, Theorem 4.2.1 sustains when $p \notin V$.

When $p \in V$, by Lemma 4.2.7, only one sensor wins. If the winning sensor is S_1, $d(p) = d_1(p) \leq d_1(p) + (\Delta_{recg} + T_b)v$. If it is S_2, there must be $d(p) = d_2(p) \leq d_1(p) + (\Delta_{recg} + T_b)v$, otherwise S_1 would have burst for T_b before the sound wave ever reaches S_2 and therefore suppressed S_2. The same reasoning applies when the winning sensor is S_3, S_4. ■

From Theorem 4.2.1, we can prove that the election delay is bounded (Corollary 4.2.3) as follows:

Proof: The winning sensor starts the burst no later than $\frac{d(p)}{v} + \Delta_{recg}$. For the square sensor grid layout, the longest bursting time is $4T_{burst}$. Hence, by time $\frac{d(p)}{v} + \Delta_{recg} + 4T_{burst} + T_b$, the winning sensor has entered the elected mode. Since the winning sensor has already stopped bursting by that time, all other sensors must have entered the *suppressed mode*. Therefore, the election process completes no later than

$$\begin{aligned} & \frac{d(p)}{v} + \Delta_{recg} + 4T_{burst} + T_b \\ \leq \quad & \frac{d_1(p)}{v} + (\Delta_{recg} + T_b) + \Delta_{recg} + 4T_{burst} + T_b \\ & \text{(Due to Theorem 4.2.1)} \\ \leq \quad & \frac{\sqrt{2}l}{2v} + 2\Delta_{recg} + 4T_{burst} + 2T_b \end{aligned}$$

That is, the election delay is upper bounded by $T^{basic}_{bound} = \frac{\sqrt{2}l}{2v} + 2\Delta_{recg} + 4T_{burst} + 2T_b$. ■

Now, let us consider how to choose an appropriate T^{basic}_{reset}. If a beep takes place at time 0, and suppose the maximal audible range of the beep is R^{max}_{beep}, by time $\frac{R^{max}_{beep}}{v} + \Delta_{recg}$, it has been recognized by the farthest away sensor. On the other hand, according to Corollary 4.2.3, every hearing node (note according to assumption A2 in Section 4.1, wireless broadcast range is at least $2R^{max}_{beep}$) either enters suppressed mode or elected mode by time T^{basic}_{bound}. Assuming a beep is recognized only when an onset is detected preceded by a period of silence, we can reset every sensor to initial listening mode at time $\max\{\frac{R^{max}_{beep}}{v} + \Delta_{recg}, T^{basic}_{bound}\}$. Therefore, as a conservative approach, it is safe to let $T^{basic}_{reset} = \max\{\frac{R^{max}_{beep}}{v} + \Delta_{recg}, T^{basic}_{bound}\}$. Hence, by time $T^{basic}_{bound} + T^{basic}_{reset}$ (called *turn-around time*) all sensors are reset to RF listening mode.

4.3 Energy-Efficient Lightning Protocol

Under Basic Lightning Protocol, a sensor keeps its radio active all the time either in RF listening mode or RF bursting mode. This can be costly in energy consumption. To save energy, a naive approach would be that a sensor turns on its RF module only when an acoustic beep is recognized. It turns off the radio after election is completed (with results sent back to the sink if necessary). However, consider the case when the acoustic beep arrives at the closest sensor S^* earlier than sensor S, such that S^* recognizes the beep, say, 11ms earlier than S. If the burst of S^* lasts for only 10ms, then S would not have turned on its RF module before S^* finishes its burst. In this case, the burst from S^* cannot suppress S. Consequently, S considers itself elected as well.

To handle the above problem and conserve energy, we propose an Energy-Efficient Lightning Protocol.

4.3.1 Protocol Details

Energy-Efficint Lightning Protocol runs the following idea: upon recognition of a beep, a sensor turns on RF and listens for a period of time before it starts to burst. Let the maximal audible radius of a beep be R_{beep}^{max}. If a sensor recognizes a beep, after $\Delta_{defer} \stackrel{def}{=} \frac{R_{beep}^{max}}{v} + \Delta_{recg}$, the beep must have reached and been recognized by the farthest sensor within the R_{beep}^{max} radius. That is, after Δ_{defer}, any sensor that can hear the beep has turned on its RF module and switched to RF listening mode. And all the rest can be the same as Basic Lightning Protocol.

The formal description of Energy-Efficient Lightning Protocol is as follows:

1. All sensors are initially in RF *sleeping mode*.

2. While in RF sleeping mode, if a beep is recognized, a sensor turns on its RF module to RF *listening mode* immediately for a duration of $\Delta_{defer} = \frac{R_{beep}^{max}}{v} + \Delta_{recg}$.

3. If no RF burst is recognized in the Δ_{defer} period, a sensor with color i enters RF *bursting mode* and transmits *without back-off* a burst of $i \cdot T_{burst}$ duration.

   Immediately after the burst, the sensor switches back to RF listening mode and samples the wireless medium for T_b. If no other RF burst is recognized, the sensor decides that it wins the election and enters the *elected mode*. Otherwise, it loses the election and enters the *suppressed mode*. In both cases, a sensor sets up a timer of length T_{reset}^{ee}.

4. At any time during RF listening mode, if an RF burst is recognized, the sensor fails the election and enters the suppressed mode. Furthermore, the sensor sets up a timer of length T_{reset}^{ee}.

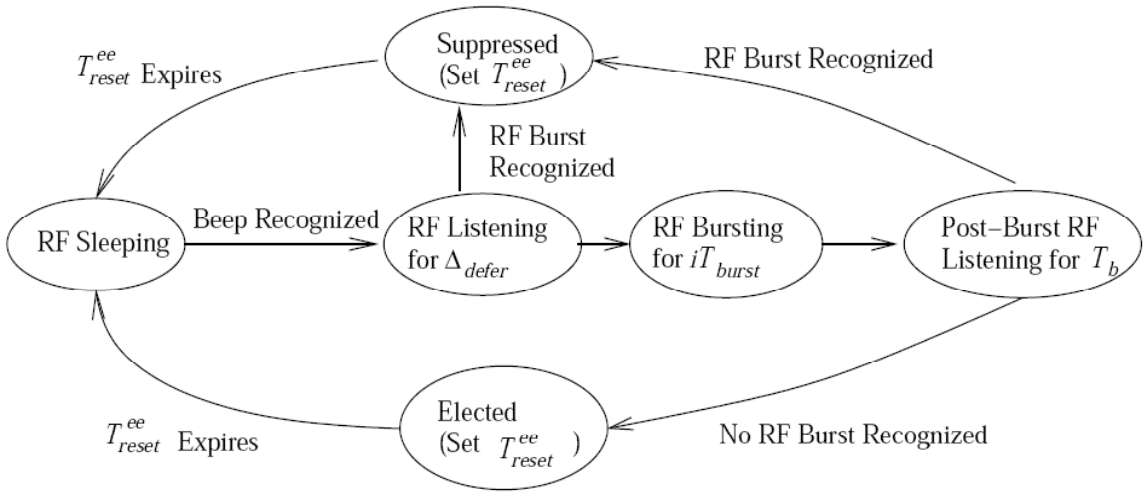


Figure 4.4: State Transition Diagram of Energy-Efficient Lightning Protocol

5. After the election is completed (i.e., after timer T_{reset}^{ee} expires), sensors return to the RF sleeping mode.

T_{reset}^{ee} is a preset constant for all sensors to reset to initial state. The setting of T_{reset}^{ee} is explained later.

Fig. 4.4 illustrates the state transition of Energy-Efficient Lightning Protocol. Compared to Fig. 4.2, an extra RF sleeping mode is introduced. In RF sleeping mode, a sensor can put its radio module to low power states to conserve energy.

4.3.2 Analysis of Energy-Efficient Lightning Protocol

1) First, we prove that Energy-Efficient Lightning Protocol preserves the desirable properties of the Basic Lightning Protocol, i.e., single winner, bounded error, $O(1)$ transmissions, and bounded election delay.

Theorem 4.3.1 (Winner Uniqueness and Error Bound) *Lemma 4.2.4∼4.2.7 and Theorem 4.2.1 remain valid for Energy-Efficient Lightning Protocol.*

Proof: According to Energy-Efficient Lightning Protocol, every sensor delays Δ_{defer} after it recognizes the beep. By the time the first RF burst starts, all sensors that can hear the beep have switched to RF listening mode. From thereon, the protocol proceeds as the Basic Lightning Protocol. In other words, it is equivalent to that the beep takes place Δ_{defer} units of time later than the actual beep with every sensor running Basic Lightning Protocol. Therefore, all proofs in Lemma 4.2.4∼4.2.7 and Theorem 4.2.1 sustain. ■

Similarly, the following corollary holds.

Corollary 4.3.2 (Election Delay Bound) *If* $T_{burst} \geq 2T_b$ *and* $l > \frac{2}{2-\sqrt{2}}(\Delta_{recg} + T_b)v$*, Energy-Efficient Lightning Protocol incurs an election delay no greater than* $T^{ee}_{bound} = T^{basic}_{bound} + \Delta_{defer} = \frac{\sqrt{2}l}{2v} + \Delta_{defer} + 2\Delta_{recg} + 4T_{burst} + 2T_b$*, where* $\Delta_{defer} = \frac{R^{max}_{beep}}{v} + \Delta_{recg}$.

Similar to the argument for T^{basic}_{reset} in Section 4.2.3, assume the maximal audible range of the beep is R^{max}_{beep}, a conservative value for the time to reset sensors to initial sleep mode from suppressed or elected mode T^{ee}_{reset} is $T^{ee}_{bound} - \Delta_{defer}$. Therefore, by time $T^{ee}_{reset} + T^{ee}_{bound}$ (called *turn-around time*), all sensors are reset to the initial RF sleeping mode.

2) Next, we evaluate the energy consumption. In the Energy-Efficient Lightning Protocol, sensors are active (with radio on) only during the election period. Since the election time is short, the energy consumption is small. For example, two AA batteries can sustain a U.C. Berkeley Mote with its radio active for 59 hours. In our implementation (Section 4.5), a burst lasts 3.2ms at the most. The RF listening time $\Delta_{defer} + T_b = 20.5$ms (suppose $R^{max}_{beep} = 20$ft). Approximately, with two AA batteries, over 9 million acoustic events can be localized using Energy-Efficient Lightning Protocol.

In what follows, unless otherwise specified, the term "Lightning Protocol" refers to Energy-Efficient Lightning Protocol.

4.3.3 A Quantitative Comparison with Data Packet (DP)-Based Scheme

Let problem scale n be the number of sensors hearing the acoustic beep. Let T^{light}_{elec} and T^{data}_{elec} be the election delay of Lightning Protocol and a DP scheme respectively. We have $T^{light}_{elec} \leq T^{ee}_{bound}$ and $T^{ee}_{bound} \sim O(1)$ (Corollary 4.3.2), therefore $T^{light}_{elec} \sim O(1)$, which is the best possible for any protocol. $T^{ee}_{bound} \sim O(1)$ for the following reasons: by the definition of T^{ee}_{bound} given in Corollary 4.3.2, the only parameter that is relevant to total number of sensors n is l, i.e. the sensor grid edge length. However, as n increases, l becomes smaller, which makes T^{ee}_{bound} smaller. One may argue that in Theorem 4.2.1,4.3.1 and Corollary 4.2.3,4.3.2, it is required that $l > \frac{2}{2-\sqrt{2}}(\Delta_{recg} + T_b)v$. This requirement is needed to guarantee only *one* sensor is elected. In cases that $l \leq \frac{2}{2-\sqrt{2}}(\Delta_{recg} + T_b)v$, there may be multiple winners within $\frac{2}{2-\sqrt{2}}(\Delta_{recg} + T_b)v$ radius (due to Theorem 4.4.1) to the sound source (there is no winner outside this circle). But in this case, with similar approach, it can be proved that both election delay and turn-around time ($T^{ee}_{bound} + T^{ee}_{reset}$) are still $O(1)$. The same conclusion also applies to Basic Lightning Protocol.

For DP schemes, sensors that hear the beep must contend for the wireless medium to broadcast their TOA readings. If an IEEE 802.11 like MAC protocol is adopted to resolve the channel contention, the expected number of data packet collisions before the *first* successful data packet broadcast grows exponentially with

the number of contending sensors [97], i.e., $\Omega(c^n)$, where c is a constant > 1. For wireless sensors equipped with half-duplex radio, once a transmission starts, it lasts till the entire packet is transmitted, and collisions are detected by lack of acknowledgments at the sender. Let T_{pack} be the time to transmit a data packet, the expected delay before the first successful data packet broadcast is, therefore, $\Omega(c^n \times T_{pack}) = \Omega(c^n)$. The expected election delay $\mathbf{E}[T_{elec}^{data}]$ to get the data packet with earliest TOA is strictly no less than $\Omega(c^n \times T_{pack})$. Therefore, $\mathbf{E}[T_{elec}^{data}] \sim \Omega(c^n)$.

One may argue that MAC contention can be alleviated by reducing the radio range in data communication. In this case, assumption A2 (see Section 4.1) that radio range is at least twice as large as the acoustic sensing range may no longer hold. Sensors overhearing the same acoustic event cannot reach each other directly. Therefore, multi-hop communications are needed to elect the closest sensors. As a result, the election delay is thus the sum of single-hop election delay and the delay incurred by multi-hop forwarding of data packets. When the sensors get denser (i.e. when n increases), the election delay increases monotonically anyway.

4.4 Random Placement of Sensors

So far, regular placement of sensors (e.g. square grid) is assumed, where nodes are colored to ensure the *uniqueness* of elected sensor. If electing multiple (closest) sensors is allowed, placement of sensors can be random, and coloring of sensors is no longer necessary. The reason is that the closest sensors burst first and suppress sensors farther away. The only modification to the Basic and Energy-Efficient Lightning Protocol is that the RF burst time of each sensor is T_{burst} instead of iT_{burst} (i is the color of the sensor). Also, as before, we require $T_{burst} \geq 2T_b$.

Let $d(S)$ be the distance from sound source to sensor S. Let S^* be the closest sensor to the sound source. Then we have:

Theorem 4.4.1 (Error Bound) *For both Basic Lightning Protocol and Energy-Efficient Lightning Protocol, with random sensor placement, only sensors within $d(S^*) + (\Delta_{recg} + T_b)v$ radius of the sound source may be elected. We denote this area as $\odot Beep$.*

Proof: First, we consider the Basic Lightning Protocol. Suppose a beep takes place at time 0. By time $\frac{d(S^*)}{v}$, the sound wave reaches the closest sensor S^* and by time $\frac{d(S^*)}{v} + \Delta_{recg}$, S^* should have recognized the beep and started bursting. By time $\tau = \frac{d(S^*)}{v} + \Delta_{recg} + T_b$, S^* should have burst for at least duration T_b. On the other hand, for any sensor S whose $d(S) > \tau v$, by time τ, it would have not yet heard the beep.

Therefore S is suppressed by S^*'s burst, i.e. any sensor whose $d(S) > \tau v = d(S^*) + (\Delta_{recg} + T_b)v$ would not win the election.

The above argument also applies to the Energy-Efficient Lightning Protocol. ■

Corollary 4.4.2 (Election Delay Bound) *If a beep takes place at time* 0*, we have (i) the election delay of Basic Lightning Protocol with random sensor placement is bounded by* $T^{basic}_{bound} = \frac{d(S^*)}{v} + 2\Delta_{recg} + T_{burst} + 2T_b$*; (ii) Energy-efficient Lightning Protocol with random sensor placement elects the closest sensors in* $T^{ee}_{bound} = \frac{d(S^*)}{v} + \Delta_{defer} + 2\Delta_{recg} + T_{burst} + 2T_b$*, where* $\Delta_{defer} = \frac{R^{max}_{beep}}{v} + \Delta_{recg}$*.*

Proof: First, we consider the Basic Lightning Protocol. By Theorem 4.4.1, only sensors within $\odot Beep$ would burst, all sensors outside of $\odot Beep$ would be suppressed by the bursts from sensors within $\odot Beep$. That is, when the sensors within $\odot Beep$ complete their Basic Lightning Protocol election, the whole election is completed.

From Theorem 4.4.1, the radius of $\odot Beep$ is $d(S^*) + (\Delta_{recg} + T_b)v$, therefore $\forall S \in \odot Beep$, the latest time that the beep would reach S is $\tau_0 = \frac{d(S^*) + (\Delta_{recg} + T_b)v}{v} = \frac{d(S^*)}{v} + \Delta_{recg} + T_b$. The latest time S would recognize the beep is $\tau_1 = \tau_0 + \Delta_{recg}$, and then it will burst for T_{burst} and perform a post-burst RF sample of T_b to see whether other RF bursts exist. Therefore, by time $\tau_2 = \tau_1 + T_{burst} + T_b = \frac{d(S^*)}{v} + 2\Delta_{recg} + T_{burst} + 2T_b$, S would have completed its election procedure.

Similar argument also applies to the Energy-Efficient Lightning Protocol. ■

Although according to Theorem 4.4.1, multiple winners may be elected, experiment results show that the number of multiple winners is still small (see Section 4.5.3). This is because of the randomness of TOA and TOA recognition time cost (t_{recg}), which equals randomly coloring sensors in some sense. As future work, we plan to further investigate randomized mechanisms to reduce the number of elected sensor within $\odot Beep$.

4.5 Experiments and Comparisons

4.5.1 Implementation and Lab Environment

We implemented Lightning Protocol on U.C. Berkeley MICA Motes, using their standard sensing boards, and an acoustic sampling rate of 8kHz. The transmission/reception of RF bursts is supported by the RF hardware of Motes [96]. The final total footprint (including TinyOS, Mote's operating system) is 5330 bytes in ROM and 187 bytes in RAM. According to our implementation, $\Delta_{recg} = 0.12$ms, $T_b = 0.4$ms and $T_{burst} = 0.8$ms.

Several experiments are conducted to evaluate Lightning Protocol and compare it against an ideal DP scheme. Video clips of the demo are available at [98].

The experiments are all conducted in a common office environment filled with daily RF interferences, such as RF interferences from IEEE 802.11a/b/g WLANs (deployed throughout the building), large-scale computer clusters and alternate current power cables (deployed right beneath the floor). The wireless medium also suffers worse large-scale path loss and multipath fading (echoes) effects than open spaces [99].

4.5.2 Experiments with Regular Sensor Placement

The experimental setup is as follows:

16 Motes are placed on square grid points, each monitoring a square area of 4ft × 4ft. Fig. 4.5 provides a top-view of the layout of sensors, sound source orientation and locations. To evaluate the robustness of the protocol, we use upright speaker oriented along the X-axis (see Fig. 4.5 and Fig. 4.6) as directional sound source. Fig. 4.6(b) demonstrates the irregular intensity field of the speaker playing a hand clap sound.

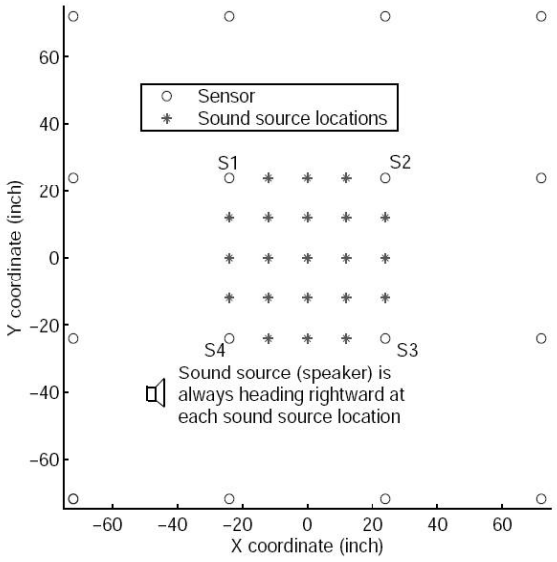
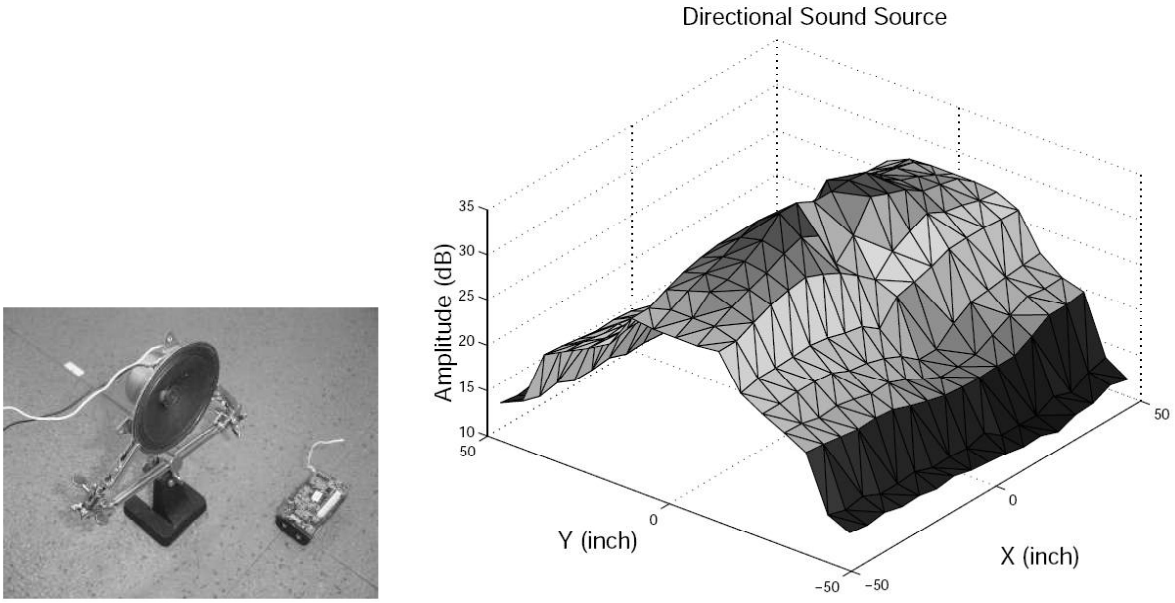


Figure 4.5: Top-view of sensor layout, sound source (speaker) orientation and locations

As a baseline, we also implement a *Data Packet* (DP) *Protocol*, as shown in Fig. 4.7. Both Lightning Protocol and DP protocol use the same TOA recognition module.

Localization Accuracy

Since the purpose of proximity-based localization is to find the closest sensor, we define the following metric to measure localization error e:



(a) (b)

Figure 4.6: Sound Source. (a) Speaker (directional sound source) and U.C. Berkeley MICA Mote; (b) Directional sound field of the speaker.

1. /* When a beep takes place */
2. All sensors report their detected TOAs;
3. Pick the sensor with the earliest TOA as the closest sensor;

Figure 4.7: Data Packet Protocol

$$e \stackrel{def}{=} dis(S) - dis(S^*), \tag{4.1}$$

where S is the location of elected sensor, S^* is the location of the sensor closest to sound source. $dis(x)$ measures the Euclidean distance between sensor x and the sound source.

For each location of the sound source in Fig. 4.5, 10 trials of experiments are carried out using Lightning Protocol and DP protocol respectively. Furthermore, the DP protocol results are ideal in the sense that we only count those trials where no data packet loss occurs. Fig. 4.8 shows localization error statistics for both schemes. From Fig. 4.8, we see that Lightning Protocol achieves comparable or even better accuracy than ideal DP protocol. This is because DP protocol requires accurate clock synchronization to determine which sensor has the earliest TOA[3] while Lightning Protocol needs no clock synchronization. [4]

[3]In our implementation, clock synchronization is done by broadcasting a sync-packet from a dedicated synchronization node before each beep. Every sensor, on receiving the sync-packet resets its local clock to 0.

[4]Some errors in our experiment may still look large. This is further explained in Section 4.6.2.

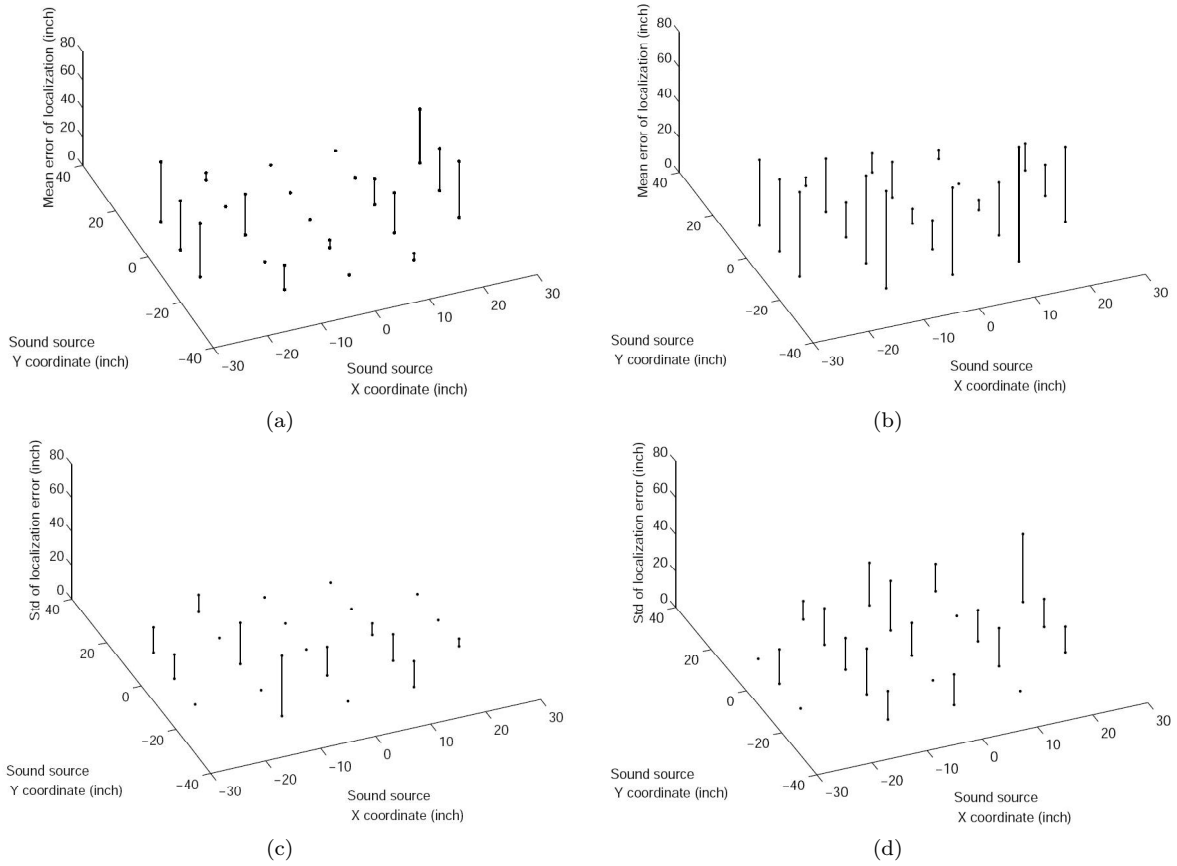


Figure 4.8: Localization Error Comparison with Regular Sensor Layout. (a) Mean of Lightning Protocol Localization Error; (b) Mean of Data Packet Protocol Localization Error; (c) Standard Deviation of Lightning Protocol Localization Error; (d) Standard Deviation of Data Packet Protocol Localization Error.

Election Delay

The election delay T_{elec}^{data} of DP protocol are determined by two factors, i.e., (*i*) the medium access control protocol used and (*ii*) the order of transmissions for sensors. In TinyOS 1.0 [95], a simple CSMA/CA mechanism with fixed contention window size cw is implemented. Upon detecting an idle channel, sensors that have backlogged packets first back-off randomly (uniformly distributed in $[0, cw]$) and then transmit if the channel remains idle. The current TinyOS implementation does not retransmit data packets in presence of collisions. We measure two delays in the DP protocol. The first one is T_{win1st}^{data}, defined as the time for the winner sensor (i.e. the sensor with the earliest TOA reading. Note the winner sensor is not necessarily the actual closest sensor.) to send out its first data packet; the second metric is T_{any1st}^{data} defined as the time it takes to send out the first data packet from any sensor. Clearly, $T_{win1st}^{data} \geq T_{any1st}^{data}$. And T_{any1st}^{data} is a lower bound of election delay for data packet based schemes.

We measure the election delay of Lightning Protocol and compare it against the delays measured in

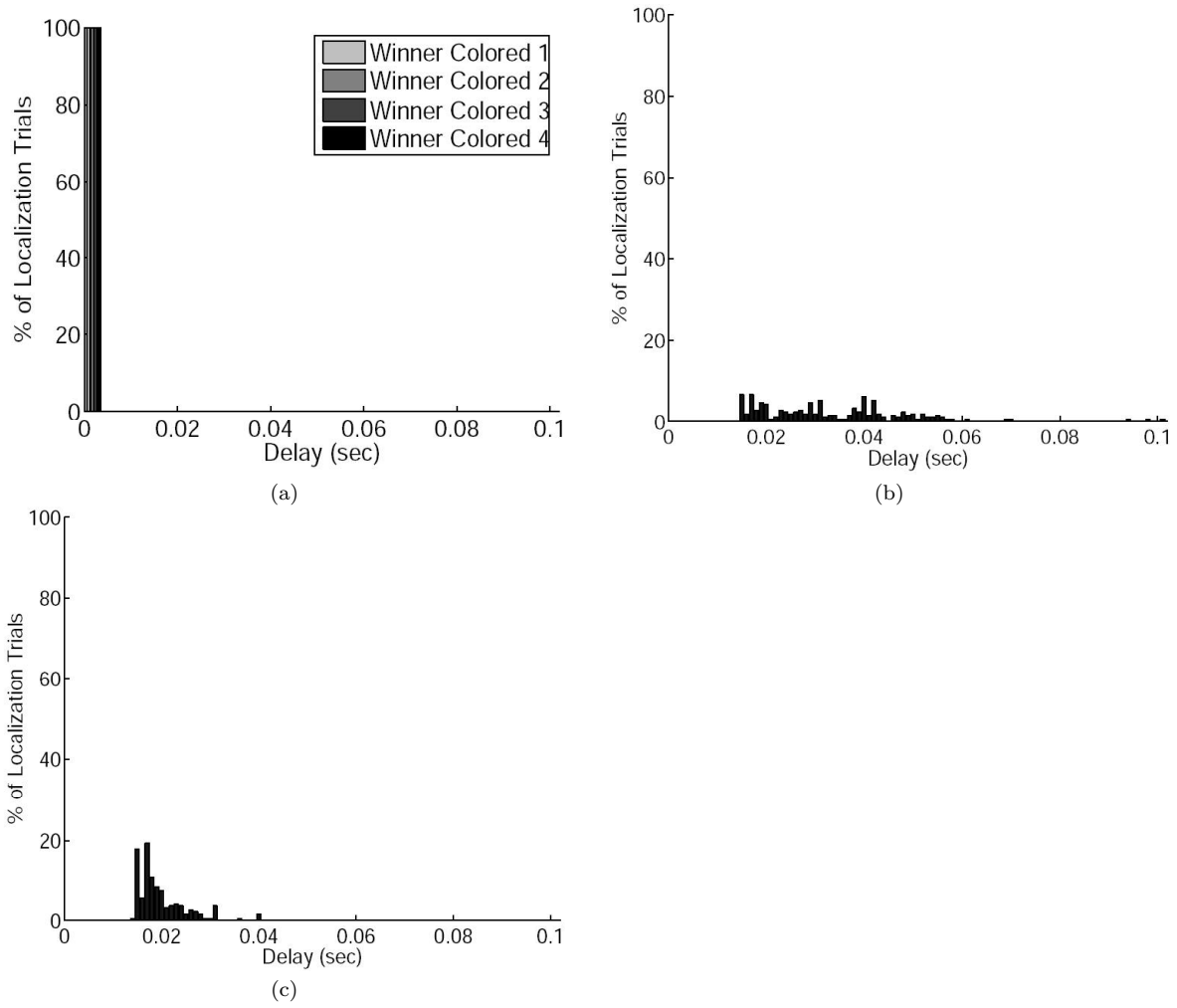


Figure 4.9: Election Delay Comparison. (a) Lightning Protocol: Election delay distribution; (b) DP Protocol: Distribution of T^{data}_{win1st} (time when the 1st packet is sent out from the winner sensor); (c) DP Protocol: Distribution of T^{data}_{any1st} (time when the 1st packet is sent out from any sensor).

DP protocol. This comparison errs on the pessimistic side for Lightning Protocol, as DP protocol does not retransmit collided packets.

Fig. 4.9(a)(b)(c) show the measurement results of T^{light}_{elec}, T^{data}_{win1st} and T^{data}_{any1st} respectively.[5] From Fig. 4.9(a), we see T^{light}_{elec} is fixed and is only determined by the color of the elected sensor; In Fig. 4.9(b) and (c), T^{data}_{win1st} and T^{data}_{any1st} are scattered because of the random back-off mechanisms in TinyOS radio stack .

Table 4.2 compares statistics of T^{light}_{elec}, T^{data}_{win1st} and T^{data}_{any1st}. Clearly, even the maximum T^{light}_{elec} (when elected sensor is colored 4) is shorter than the minimum T^{data}_{win1st} and T^{data}_{any1st}. This indicates that Lightning

[5]The comparison is made between Basic Lightning Protocol and DP Protocol. Both protocols share the same acoustic propagation delay from sound source to the closest sensor, and share the same TOA recognition module, therefore time zero refers to the time that TOA is recognized at the closest sensor. For Energy-Efficient Lightning Protocol, there should be an additional Δ_{defer} delay. However, if the corresponding DP Protocol is also energy efficient, it should also have the additional Δ_{defer} delay.

Table 4.2: Statistics of Election Delay

Metric(ms)	Max	Min	Mean	Standard Deviation
T_{elec}^{light}, winner colored 4	3.6	3.6	3.6	0
T_{win1st}^{data}	101.8	15.4	33.6	14.9
T_{any1st}^{data}	40.8	14.9	20.2	5

Protocol always incurs less election delay than any data packet-based schemes.

Number of Transmissions

Next, we count the number of transmissions in the Lightning Protocol. From the analysis in Section 4.2.3, we know that in Lightning Protocol, at most 4 sensors would burst and each bursts only once, independent of the total number of sensors hearing the beep. This is proved by the experimental results: 81.4% of the localization trials only involves *one* burst (broadcast); 18.6% of the trials incur two bursts; and none involves more than two bursts.

4.5.3 Experiments with Random Sensor Placement

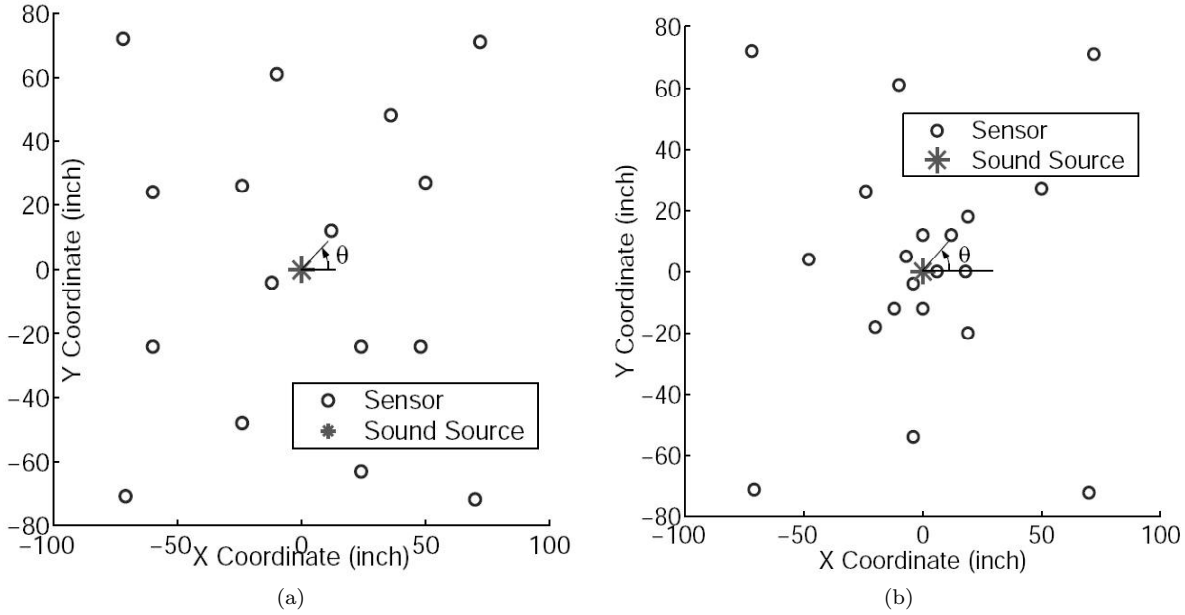


Figure 4.10: Random Sensor Layout Testbed Settings. (a) Uniform Random Layout; (b) Clustered Random Layout with Additional Sensors Around the Sound Source.

We use the same sound source used in Section 4.5.2, but deploy sensors in random layouts, as shown in Fig. 4.10. The random layout in Fig. 4.10(a) is *uniform*, where 16 sensors are uniformly distributed in a square area of 12ft × 12ft. On the other hand, the random layout in Fig. 4.10(b) is *clustered*, where 20

sensors are randomly distributed but deliberately made denser around the sound source, so as to intensify election contention within $\odot Beep$ (see Theorem 4.4.1 and Corollary 4.4.2). For random layout, it is not supposed to elect only *one* winner, therefore sensor coloring is no longer necessary. In practice, all sensors are colored 1, so that each sensor's bursting time is T_{burst}, if there is any.

It is infeasible to test every random layout, therefore Monte Carlo is carried out. We utilize the feature that the sound source (speaker) has a irregular sound intensity field (see Fig. 4.6(b)). For each layout of Fig. 4.10(a) and (b), eight speaker orientations are tried (as shown in Fig. 4.10(a) and (b), the speaker orientation $\theta = 0^o, 45^o, 90^o, 135^o, 180^o, 225^o, 270^o$ and 315^o), which equals testing eight different random sensor layouts. For each speaker orientation, 30 localization trials are carried out.

Localization Accuracy

As mentioned earlier, for both layouts in Fig. 4.10(a) and (b), and for each speaker orientation, 30 trials are carried out using Lightning Protocol and another 30 trials are carried out using ideal DP protocol. We use the same metric as defined in Equation (4.1) (Section 4.5.2) to measure the accuracy of Lightning Protocol, and compare it against the results of DP Protocol. According to Theorem 4.4.1, Lightning Protocol may result in more than one winning sensors. In this case, we pessimistically count the sensor farthest away from the sound source as the final winning sensor.

Fig. 4.11(a.1) and (a.2) shows the localization error statistics of Lightning Protocol and DP protocol respectively with the uniform random layout testbed (see Fig. 4.10(a)). Fig. 4.11(b.1) and (b.2) shows the statistics for the clustered random layout testbed (see Fig. 4.10(b)). From these figures we observe that Lightning Protocol achieves comparable, or even better localization accuracy than DP Protocol. This holds even when there are denser sensors around the sound source, which tends to increase the number of multiple winners of the Lightning Protocol election. The better accuracy of Lightning Protocol is due to its self-synchronization property.

Election Delay

We also compare the election delays of Lightning Protocol (T_{elec}^{light}) with T_{win1st}^{data} and T_{any1st}^{data} of DP Protocol, where T_{win1st}^{data} is the time cost to send out the first packet from the winner sensor [6], and T_{any1st}^{data} is the time cost to send out the first packet from any sensor.

Experiment results from both the uniform and clustered random layout testbeds are shown in Fig. 4.12. Table 4.3 compares statistics of T_{elec}^{light}, T_{win1st}^{data} and T_{any1st}^{data}. Clearly, in both random layouts, T_{elec}^{light} is much

[6] for DP Protocol, there is always only one winner, given that ties are broken randomly.

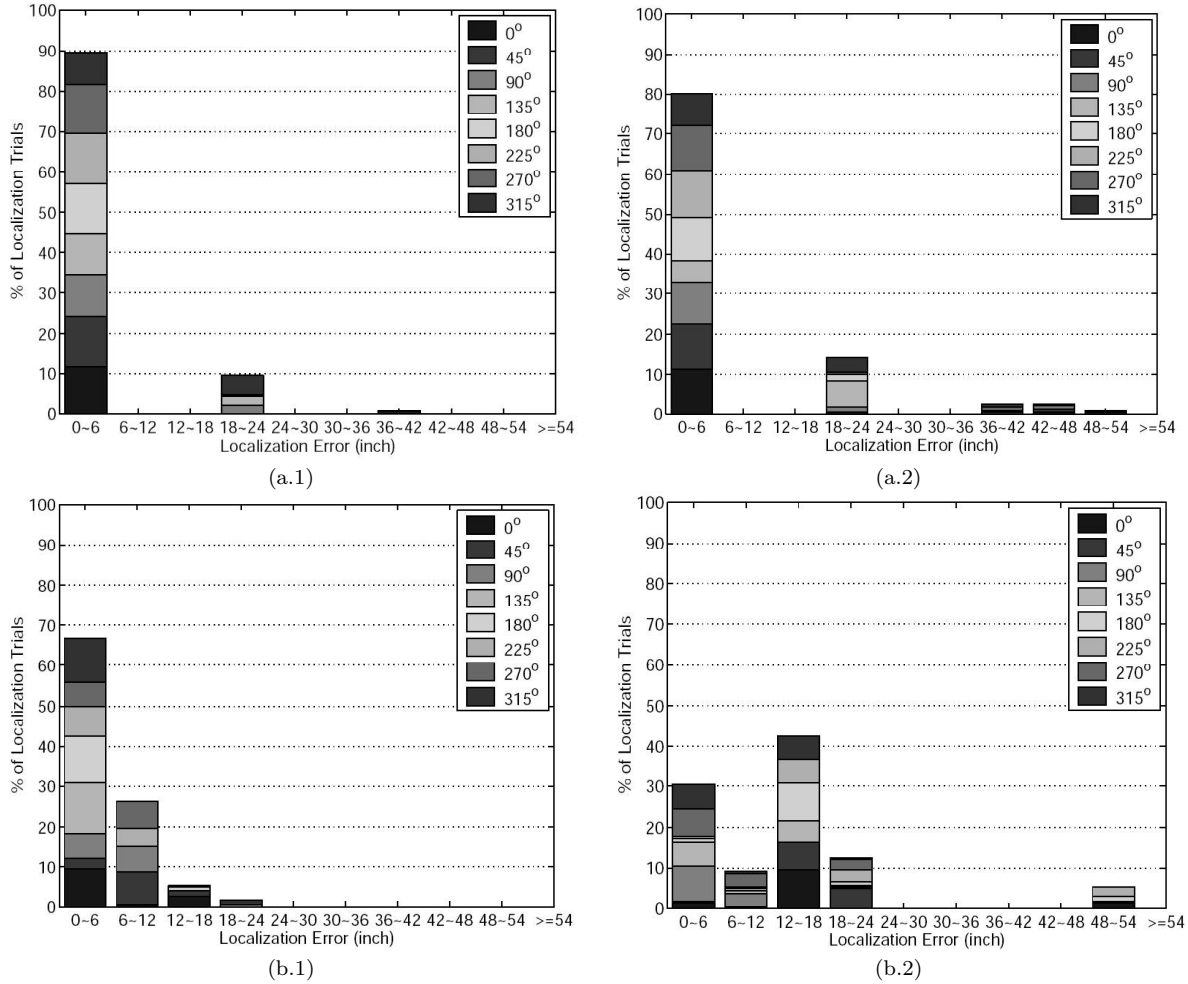


Figure 4.11: Localization Error Comparison with Random Sensor Layout. (a.1) Lightning Protocol with Uniform Random Layout; (a.2) DP Protocol with Uniform Random Layout; (b.1) Lightning Protocol with Clustered Random Layout; (b.2) DP Protocol with Clustered Random Layout.

less than the minimal T^{data}_{win1st} and T^{data}_{any1st}. This indicates that Lightning Protocol, under random layout, still incurs less election delay than any data packet-based schemes.

Number of Transmissions and Winners

From Section 4.4, we know that owing to the lack of regular layout and coloring, the random layout Lightning Protocol is less efficient in suppressing sensors from contending (bursting) and winning the election. However, experiment results show that the degradation is not significant. Fig. 4.13(a) and (b) illustrates the statistics of number of bursts per localization trial for the uniform and clustered random layouts respectively. Note that in the clustered random layout, there are more sensors around the sound source. This would further impair the efficiency of burst suppression. However, as observed from the results of Fig. 4.13(a) and (b), the

Table 4.3: Statistics of Election Delay with Random Sensor Layouts

Layout	Metric(ms)	Max	Min	Mean	Standard Deviation
Uniform	T_{elec}^{light}	1.2	1.2	1.2	0
	T_{win1st}^{data}	123.5	15.6	47.3	22.7
	T_{any1st}^{data}	45.7	15.6	18.6	4.1
Clustered	T_{elec}^{light}	1.2	1.2	1.2	0
	T_{win1st}^{data}	99.4	15.6	40.5	18.7
	T_{any1st}^{data}	43.0	15.6	18.7	4.6

total number of bursts remains low.

Fig. 4.14 show the statistics on the number of winners: the majority of the localization trials only elect *one* winner. The percentage of having 2 or 3 winners is low, and there is no trial that elects more than 3 winners.

That Lightning Protocol performs well with random layouts and single coloring can be attributed to the randomness of TOA recognition. Firstly, because of the random layout, the acoustic wave propagation delay to each sensor is randomized. In addition, the TOA is not immediately recognized when the acoustic wave physically reaches a sensor. Instead, there is a random recognition time $t_{recg} \in [0, \Delta_{recg}]$. Both factors contribute to differentiating the time when the bursts start and end. Therefore, majority of redundant RF bursts are still suppressed.

4.6 Discussion

4.6.1 Multiple Acoustic Events

Lightning Protocol can effectively handle multiple acoustic events (beeps) if the events are separated either temporally by at least $T_{bound}^{\star}+T_{reset}^{\star}$ units of time or spatially by at least 4 times the maximal acoustic audible range (i.e. $4R_{beep}^{max}$, where R_{beep}^{max} is the maximal audible radius from any sound source). The minimal spatial separation requirement is derived as follows (Fig. 4.15). Suppose beep b_1 and b_2 take place simultaneously, and b_1 and b_2 are $4R_{beep}^{max}$ apart from each other. The most remote sensors that can hear b_1 and b_2 are R_{beep}^{max} away from b_1 and b_2 respectively (denoted as S_1 and S_2 in the Fig. 4.15). S_2's RF broadcast only needs to cover a radius of $2R_{beep}^{max}$ to reach all sensors that can hear b_2. Therefore, the broadcast of S_2 does not interfere with any sensor (say S_1) that can hear b_1, and vise versa.

Take our experiment testbed settings for example, we assume $R_{beep}^{max} = 20$ft. this should translate to a separation of beeps by at least 34.3ms in time *or* by at least 80ft in space.

4.6.2 Violation of Theoretical Assumptions on Regular Layout

Fig. 4.8 shows both Lightning and DP protocols may have large error when the sound source is at $(\pm 24, 12)$, $(\pm 24, 0)$ and $(\pm 24, -12)$. This is because our sound source, a directional speaker shown in Fig. 4.6(a), does *not* fully comply with the bounded directionality assumption(see Section 4.1 P2), which says the closest sensor shall always be able to recognize the beep no matter what direction it is oriented toward the sound source. When the speaker is at $(\pm 24, 12)$, $(\pm 24, 0)$ and $(\pm 24, -12)$, the nearest sensor is exactly at the speaker's right/left side (see Fig. 4.5), where the sound intensity may be too weak, which sometimes causes the sensor's simple threshold-based algorithm fail to detect the beep. Also, as an early generation of Motes, the MICA Motes' microphones may occasionally fail to detect the beep, adding another source of error.

Fortunately, Lightning Protocol still works when some sensors fail to detect the beep. Under such cases, the sensor layouts can be regarded as random layouts, which still has good properties. Note when the random layout Lightning Protocol elects multiple winners, we count the winner farthest away from the sound source for localization error. Of all the trials for Fig. 4.8, we observe only 5.7% trials generate two winners; and no trial generates more than two winners.

4.6.3 Multipath Effects

As for wireless multipath, in indoor environments, the common multipath spread is in the order of 0.0001ms. In our Lightning Protocol implementation, $T_{burst} = 0.8$ms, $T_b = 0.4$ms, which are way larger than 0.0001ms. This means the wireless channel is flat, and the RF multipath effect can be neglected [99].

As for acoustic multipath, this paper assumes open space or lightly obstructed environment where acoustic multipath effects is negligible (see Section 4.1 A4). In practice, our experiments show that Lightning Protocol can tolerate acoustic multipath effects in a common office environment satisfactorily. Lightning Protocol may even handle worse acoustic multipath effects with extra time cost: First, dense deployment of sensors empirically guarantees LOS between the sound source and its closest sensor. Therefore the closest sensor still bursts first. Second, Δ_{defer} is increased in case some remote sensor does not have LOS to the sound source but can hear the beep arrived through reflected path. Third, $T^{\star}_{reset}$ is increased to prevent sensors from resetting too early to mistake echoes of the beep as new beeps.

4.7 Related Work

Existing solutions to acoustic localization using wireless sensors mainly fall into two categories.

The first category requires the use of some high-capability nodes to conduct triangulation based on

readings gathered from a larger population of low-end sensors [91, 92, 100–106]. In Sheng et al. [100] and Aslam et al. [101], recursive particle-filtering algorithms are devised to asymptotically converge to a moving target's track over time. Though convergence is proven, there is no hard time bound on convergence time. In [91,92,102,103], *Maximum Likelihood*(ML)-based localization methods are proposed based on intensity or TOA readings of an array of sensors. In Kim et al. [102], binary readings are aggregated at a central tracking node where regressions are carried out to best fit the weighted set of readings. In Sheng et al. [91], sound sources are assumed to be omni-directional and the attenuation model is known. Each sensor estimates its distance to sound source based on the detected acoustic intensity. In Wang et al. [92], each sensor reports its local TOA, and the algorithm scans the whole monitored area to find the grid point that best matches the TOA readings of the sensor array. Simon et al. [103] implement a sniper gun localization system based on TOA of muzzle blast using Maximum Likelihood estimation on consistent TOA readings. Both VigilNet [104] and ExScal [105,106] systems focus on providing holistic middleware architecture solution for moving target tracking. In VigilNet, the target localization module uses magnetic sensors. Every sensor detecting a target within its sensing range reports its coordinates and timestamp. The group leader uses regression to estimate the most likely track of the moving target. In ExScal, all sensors that detect a target within its sensing range report its coordinates and timestamp. For each time window, the leader calculates the centroid of the convex region that envelops all sensors currently detecting the target as the target's location; and uses correlation over successive time windows to estimate the track of the target.

The second category of acoustic localization schemes is based on *proximity-based localization* among homogeneous wireless sensor nodes [107–110]. In Liu et al. [107] and Blum et al. [108], all sensors detecting an impulsive sound exchange their sound intensity readings or TOAs using multi-hop data communication. The sensor with the best reading wins the election. Chen et al. [109] devise a back-off based method to accelerate the election process. This scheme works well if the sound sources are omni-directional and of known intensity. Oh et al. [110] propose using Viterbi algorithm and hidden Markovian model to track targets in a sparse wireless sensor network, where sensors' sensory coverages are non-overlapping. This algorithm is mainly designed for tracking. For locating individual acoustic events, it would just give the singular sensor that covers the event's location, as sensors' sensory coverages are non-overlapping. Generally speaking, with densely deployed sensors, proximity-based localization achieves acceptable accuracy. More importantly, since proximity-based localization requires only algorithmic comparisons, it is lightweight and can be carried out using only low-end micro-sensors. A proximity-based localization scheme can also serve the purpose of leader election, which is important for activities such as dynamically collaborating sensors around the sound source to achieve higher localization accuracy.

Compared to all the aforementioned approaches, we claim Lightning Protocol is the first scheme that provides a hard real-time ($O(1)$) guarantee on acoustic event localization, and it only involves a few most basic computations (specifically, a few algorithmic comparison operations). The simplicity of Lightning Protocol allows it being implemented on very cheap low-end wireless sensors. The acoustic localization scheme is based on TOA, but unlike conventional TOA based localization schemes which need clock synchronization (simple schemes may need coarse-grained clock synchronization as those deployed in S-MAC [111] and Z-MAC [112]; sophisticated schemes [92] may need microsecond level clock synchronization, such as RBS [113]), Lightning Protocol eliminates the need of clock synchronization among sensors. Lightning Protocol is also robust to (bounded) directionality of sound sources and variation of sensor density.

The content of this chapter is published in [114] and [115].

4.8 Conclusion

In this paper, we exploit the fact that electromagnetic waves propagate much faster than acoustic waves to devise Lightning Protocol, which elects the closest sensor to an acoustic event in a network of low-end wireless sensors. This protocol can be used for proximity-based localization or leader election for sensor collaboration. Both theoretical analysis and experimental results are presented. Lighting Protocol is shown to have a very short and bounded delay ($O(1)$). It only incurs $O(1)$ wireless broadcasts. The majority (81.4% with regular sensor layout and $> 75\%$ with random sensor layout) of our experimental trials incur only one wireless broadcast. The protocol does not involve wireless data packet communication, instead, it deploys RF bursts so that concurrent overlapping wireless broadcasts are allowed. This greatly simplifies the design of its wireless communication module, and makes it faster and more reliable. Moreover, the protocol has little computation and storage complexity, and does not require clock synchronization among distributed sensors. Our experiments demonstrate that the accuracy of Lightning Protocol is comparable to and often better than an ideal DP scheme. The protocol is energy-efficient in the sense that sensor nodes only turn on radio in an on-demand fashion in presence of acoustic events for a constant bounded time. Finally, we demonstrate through experiments that Lightning Protocol can handle directional sound sources with variable intensities, and is empirically feasible in a common office environment.

In our future work, we plan to investigate more effective methods to handle multiple sound sources.

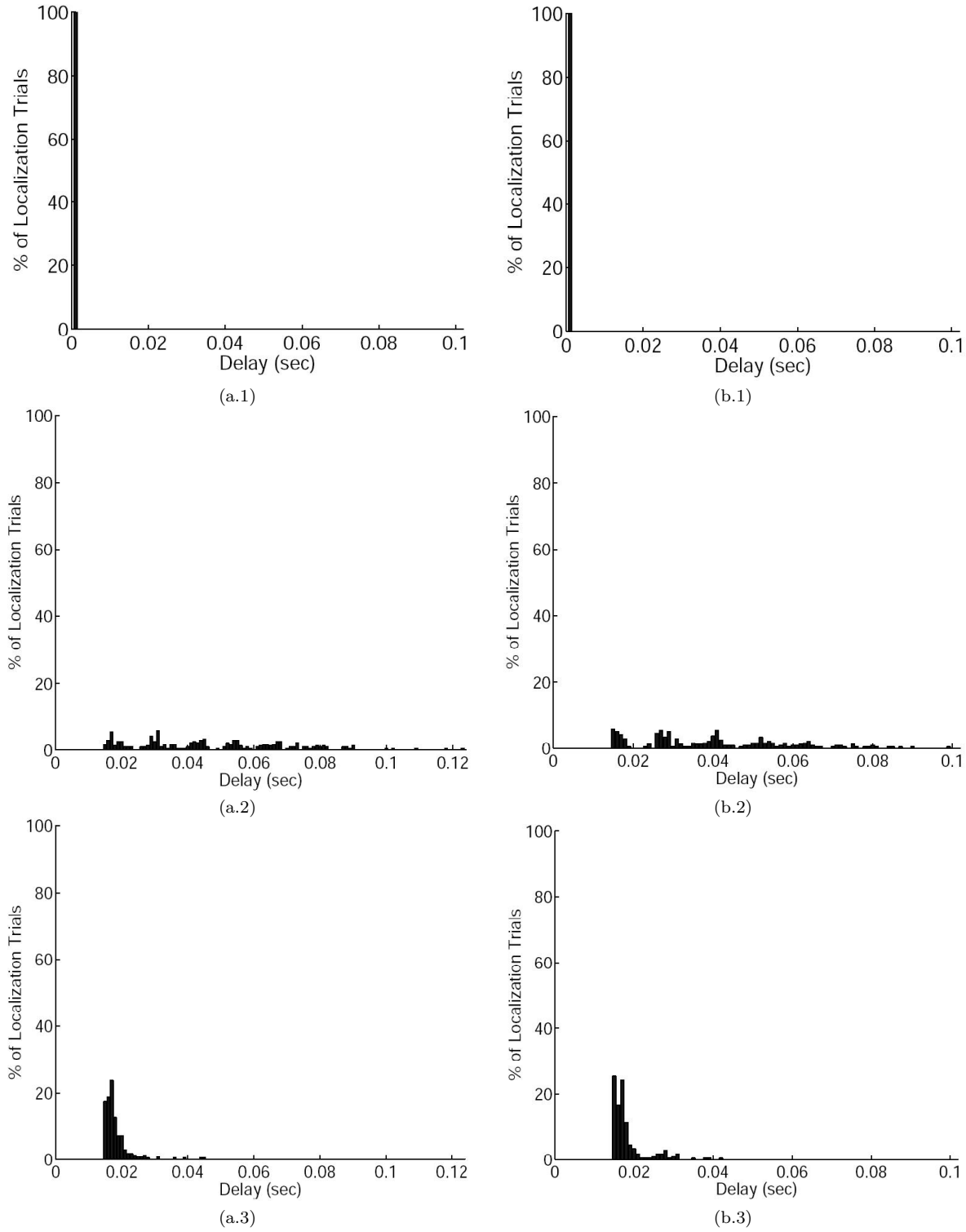


Figure 4.12: Election Delay Comparison with Random Sensor Layouts. (a.1) Lightning Protocol with Uniform Random Layout; (a.2) DP Protocol with Uniform Random Layout: Distribution of T^{data}_{win1st}, i.e. time when the 1st packet is sent out from the winner sensor; (a.3) DP Protocol with Uniform Random Layout: Distribution of T^{data}_{any1st}, i.e. time when the 1st packet is sent out from any sensor; (b.1) Lightning Protocol with Clustered Random Layout; (b.2) DP Protocol with Clustered Random Layout: Distribution of T^{data}_{win1st}, i.e. time when the 1st packet is sent out from the winner sensor; (b.3) DP Protocol with Clustered Random Layout: Distribution of T^{data}_{any1st}, i.e. time when the 1st packet is sent out from any sensor.

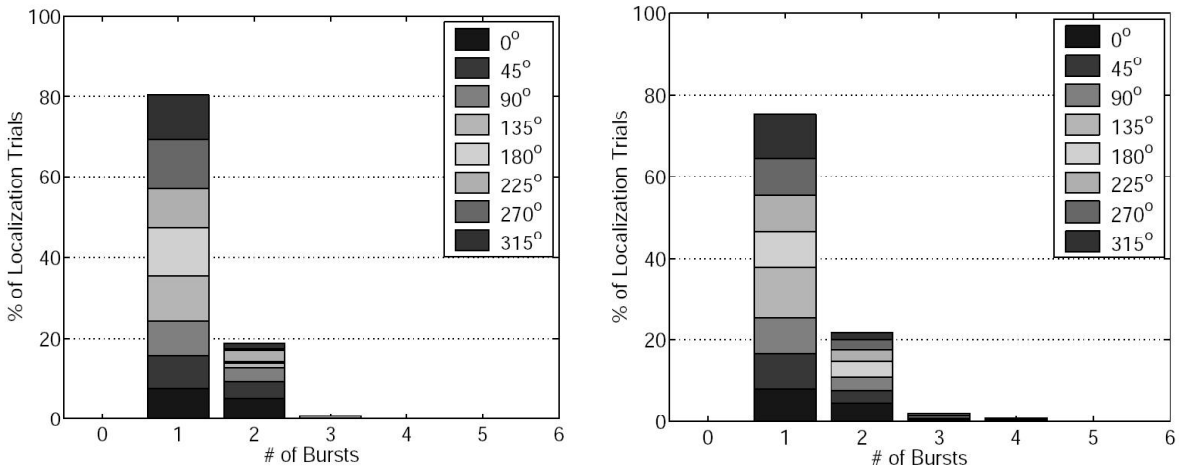


Figure 4.13: Number of Bursts per Localization Trial. (a) Lightning Protocol with Uniform Random Layout; (b) Lightning Protocol with Clustered Random Layout.

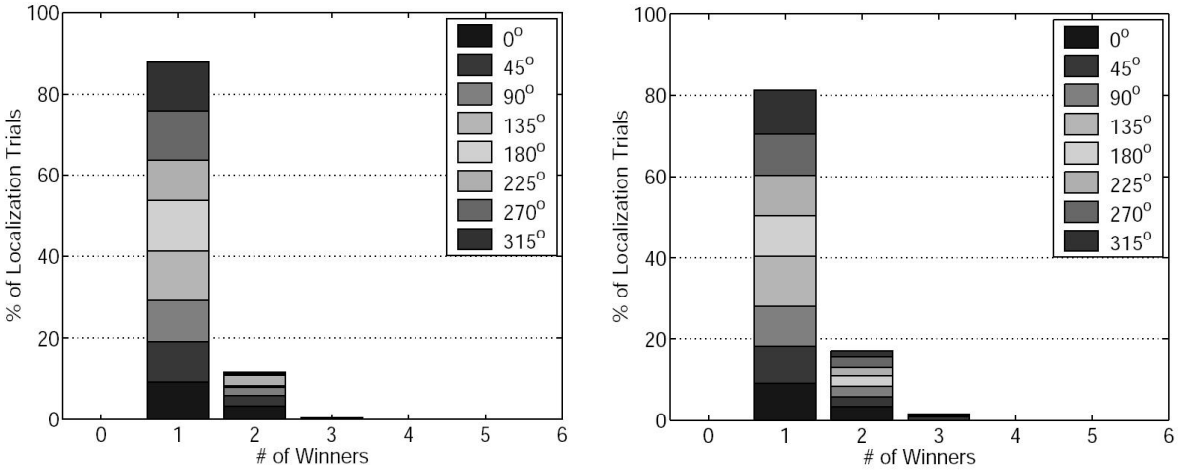


Figure 4.14: Number of Winners per Localization Trial. (a) Lightning Protocol with Uniform Random Layout; (b) Lightning Protocol with Clustered Random Layout.

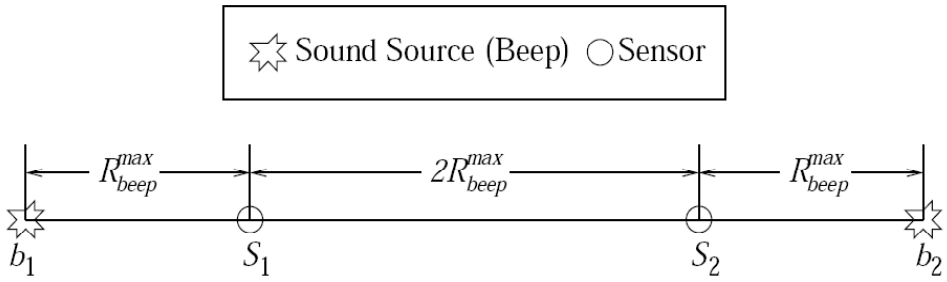


Figure 4.15: Sufficient spatial separation of two simultaneous beeps

Chapter 5

CONCLUSION AND FUTURE WORK

By converging the cyber world and the physical world, CPS is widely expected to create unprecedented impacts on computer science and the society. In fact, in the President's Council of Advisors on Science and Technology (PCAST)'s 2007 report, CPS is listed as the top of eight strategic directions to guarantee the United State's technological leadership in the world.

The many CPS applications calls for a large variety of building blocks to lay the infrastructural fundation. One big category of these building blocks are the real-time components.

For example, we need a combined wireless and wired real-time network infrastructure to support telepresence, next generation industrial control, and MDPnP. We want real-time wireless LANs at the last hop for better mobility and flexibility; and want real-time wired WAN as the backbone for reliability, throughput, and legacy reuse considerations. For real-time wireless LAN, the top concern resides in communication reliability/robustness. On the one hand, wireless communication by nature is less reliable / robust than wired communication due to large-scale path loss, multi-path, and *Radio Frequency* (RF) jamming. Many CPS environments, such as hospitals, underground mines, and factories deteriorate the situation: heavy obstructions increase large-scale path loss and multi-path; *Electro-Magnetic Interference* (EMI) from electric motors, welding, and power cords persistently jam the wireless channel; and non-cooperative RF devices can be accidentally or macliciously turned on to create additional jamming. On the other hand, real-time requirements mendate wireless connections to work continuously even under adverse channel conditions; backoff based MAC is not allowed. For wired WAN backbone, the main challenge lies in the switch (a.k.a. router) design for efficient real-time packet forwarding. The success of such design largely depends on its backward compatibility with nowadays main stream switch designs, which, however, are not for real-time.

As another example, we need real-time acoustic event localization services in the wireless sensor networks, which is a key infrastructure for many CPS applications, such as assisted living, environment monitoring, and security. Yet how to design a hard real-time, fast, and lightweight acoustic event localization mechanism for low-end wireless sensors is another challenge.

Efforts are made in the previous chapters to address these challenges.

In Chapter 2, we compare various wireless communication paradigms and show that by deploying largest processing gain possible (i.e. lowest bit rate), DSSS-CDMA cell phone paradigm (i.e. each control loop occupies a CDMA channel between the remote station and the base station) achieves much higher reliability/robustness than main stream alternatives, such as IEEE 802.11, IEEE 802.15.4, and convolutional coding.

In Chapter 3, we propose a wired WAN real-time switch design, which is not only compatible to, but also simpler than the main stream *i*SLIP [13] [14] switch design. Specifically, since real-time flows are predetermined and run consistantly for long time, an *i*SLIP switch can be easily revised into a real-time switch: we only need to grant according to a deterministic schedule; and eliminate the request and accept steps. The schedule can be represented as an N by M matrix S, where N is the number of input(output) ports of the crossbar switch, and M cell-time is the period of the schedule. Let $S(i,k)$ denote the element at the ith row and kth column of S. Then at the kth cell-time of each period, the ith input port forwards a cell to the $S(i,k)$th output port. Basically, this means the switch serves each real-time flow with clock-driven time-slicing. A flow of period P cell-time and message size of C cells are over-provisioned with a server that forwards $\lceil C/\lfloor P/M \rfloor \rceil$ cells every M cell-time. By mapping each real-time flow to a server, we derive an $N \times N$ demand matrix D. The element at the ith row and jth column of D, denoted as $D(i,j)$, means every M cell-time, input port i has to forward $D(i,j)$ cells to output j. We prove that as long as each input (ouput) is required to forward (receive) no more than M cells per M cell-time period, demand matrix D can always be scheduled within polynomial time ($O(N^4)$).

In Chapter 4, we propose a hard real-time, fast, and lightweight acoustic event localization protocol, the Lightning Protocol, which elects the closest sensor to an acoustic event with only $O(1)$ RF broadcasts. Basically, wireless sensors are deployed in a square grid pattern. Every sensor is colored i ($i = 1, 2, 3, 4$), so that for any point on the plane, the enclosing four sensors have distinct colors. A sensor is either in RF listening or broadcasting mode (never both). Whenever a free (i.e. its state is neither winner nor loser) sensor hears the acoustic event, it broadcasts a noise on the wireless carrier for i time units. After that, if it does not hear any other noise on the wireless carrier, it wins the election; otherwise it loses the election. Whenever a free sensor hears a noise on the wireless medium, it loses the election. We prove this protocol elects the closest sensor with only $O(1)$ RF broadcast within $O(1)$ time. We also designed Energy Efficient Lightning Protocol which only turns on RF module during localization period. Experiments using U. C. Berkeley Mica Motes show the feasibility of the protocol in lab environments.

My future work shall focus on integrating all the aforementioned building blocks to provide a comprehensive platform for supporting various CPS applications. As an initial attempt, I choose assisted living

as a background application, and build the I-Living [116] [117][1] open architecture. The reason of choosing assisted living is two folded. On the one hand, assisted living has important social and economic impact. On the other hand, assisted living's demand for system integration is compelling: it involves a myriad of real-time/embedded building blocks and applications, including wireless communication, tele-presence, localization etc.

According to the I-Living project experience, system integration stands out as the most eminent challenge. Many factors can conflict, such as the wireless communication schemes, real-time/embedded devices' behaviors, systems configurations, OS libraries etc. Just to give an example: to support mobility, we decide to use the cell phone as the middleware platform for integrating multiple assisted living applications. The cell phone only supports Windows Mobile and Embedded Linux. For weighing application, which is an important assisted living application, the device that we have requires using Avetana JSR-82 library. However, the Avetana JSR-82 library only runs on full-fledged Windows and Linux, not Windows Mobile or Embedded Linux. Hence finally, we have to give up the cell phone, and use laptop instead as the middleware platform, just to run full-fleged Windows/Linux.

To deal with the system integration challenge, we need input from various computer science disciplines in addition to real-time systems. We need formal methods, model checking to verify the consistancy of design, assumptions, and requirements. We need programming language to facilitate the programming and description of assumptions. We need data mining to discover the normal and abnormal behaviors of the interactions between embedded systems etc. System integration will create a huge problem space will many valueable research topics.

[1] In [117], the "I-Living" open architecture is named "PAS" open architecture.

References

[1] "Working group summary: Critical physical infrastructure," *NSF Cyber-Physical Systems Workshop*, October 2006.

[2] R. Gupta and K. G. Shin, "Working group summary: Infrastructure and building blocks," *NSF Cyber-Physical Systems Workshop*, October 2006.

[3] "Working group summary: Scientific foundations and education," *NSF Cyber-Physical Systems Workshop*, October 2006.

[4] M. Spong and K. Nahrstedt, "Working group summary break-out session on tele-interaction," *NSF Cyber-Physical Systems Workshop*, October 2006.

[5] B. Walsh, "(Global Warming) 51 Things We Can Do to Save the Environment: 14. Ride the Bus," *TIME (U.S. Edition)*, vol. 169, Apr. 2007.

[6] `http://cta.ornl.gov/data/new_for_edition26.shtml`.

[7] `http://www.infinibandta.org`.

[8] `http://mdpnp.org`.

[9] J. M. Goldman, "Medical device connectivit for improving safety and efficiency," *American Society of Anesthesiologists Newsletter*, vol. 70, May 2006.

[10] Alliance for Aging Research, *Ten reasons why America is not ready for the coming age boom.* `http://www.agingresearch.org/brochures/nevernever/nevernever.pdf`, Spring 2002.

[11] Berenson RA, Urban Institute, "Public policy lecture: Quality, chronic care, and developments in physician payments," *Presented Am Geriatric Soc Annual Meeting*, 2006.

[12] M. Magee, *Health Politics: Power Populism and Health.* Bronxville, NY: Spencer Books, 2005.

[13] N. McKeown, "The *i*SLIP scheduling algorithm for input-queued switches," *IEEE/ACM Transactions on Networking*, vol. 7, Apr. 1999.

[14] N. McKeown, M. Izzard, A. Mekkittikul, B. Ellersick, and M. Horowitz, "The tiny tera: A packet switch core," *IEEE Micro*, pp. 26–33, Jan/Feb 1997.

[15] Q. Wang, X. Liu, W. Chen, L. Sha, and M. Caccamo, "Building robust wireless LAN for industrial control with the DSSS-CDMA cell phone network paradigm," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 706–719, June 2007.

[16] F. D. Pellegrini, D. Miorandi, S. Vitturi, and A. Zanella, "On the use of wireless networks at low level of factory automation systems," *IEEE Transactions on Industrial Informatics*, vol. 2, May 2006.

[17] A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proceedings of the IEEE (Invited Paper)*, vol. 93, pp. 1130–1151, June 2005.

[18] N. J. Ploplys, P. A. Kawka, and A. G. Alleyne, "Closed-loop control over wireless networks," *IEEE Control Systems Magazine*, vol. 24, pp. 58–71, June 2004.

[19] *IEEE Transactions on Industrial Informatics – Special Issue on Wireless Factory and Industrial Automation Part I*, vol. 3. May 2007.

[20] *IEEE Transactions on Industrial Informatics – Special Issue on Wireless Factory and Industrial Automation Part II*, vol. 3. Aug. 2007.

[21] H. Ye and G. Walsh, "Real-time mixed-traffic wireless networks," *IEEE Trans. Ind. Electron.*, vol. 48, no. 5, 2001.

[22] H. Ye, G. Walsh, and L. Bushnell, "Wireless local area networks in the manufacturing industry," in *Proc. American Control Conf.*, 2000, pp. 2363–2367.

[23] S. Cavalieri and D. Panno, "A novel solution to interconnect fieldbus systems using IEEE wireless LAN technology," *Comput. Standards Interfaces*, vol. 20, no. 1, pp. 9–23, 1998.

[24] S. Jiang, "Wireless communications and a priority access protocol for multiple mobile terminals in factory automation," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 137–143, 1998.

[25] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall, second ed., 2002.

[26] F. Zhang, "Investigation of electromagnetic interference of pwm motor drives in automotive electrical systems," Tech. Rep. TR-99-004, MIT Laboratory for Electromagnetic and Electronic Systems, 1999.

[27] G. Antonini, S. Cristina, *et al.*, "A prediction model for electromagnetic interferences radiated by an industrial power drive system," *IEEE Transactions on Industry Applications*, vol. 35, Jul/Aug 1999.

[28] D. Seto and L. Sha, "A case study on analytiical analysis of the inverted pendulum real-time control system," Tech. Rep. CMU/SEI-99-TR-023, CMU SEI, 1999.

[29] E. A. Parr, *Industrial Control Handbook*. Industrial Press, third ed., 1999.

[30] R. B. Kieburtz, "The step motor - the next advance in control systems," *IEEE Transactions on Automatic Control*, vol. 9, no. 1, 1964.

[31] B. Kuo, *Theory and Applications of Step Motors*. West Publishing Company, 1974.

[32] *Stepper Motor System Basics*. `http://www.ams2000.com/stepping101.html`.

[33] *ZigBee Alliance*. `http://www.zigbee.org/en/index.asp`.

[34] *IEEE Standard 802.11*. 1997.

[35] *IEEE Standard 802.15.4*. 2003.

[36] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communication*. Prentice Hall, Apr. 1995.

[37] M. K. Simon, J. K. Omura, *et al.*, *Spread Spectrum Communications Handbook, Electronic Edition*. McGraw-Hill, 2002.

[38] R. C. Dixon, *Spread Spectrum Systems with Commercial Applications*. Wiley-Interscience, Apr. 1994.

[39] S. Haykin, *Communications Systems*. Wiley, third ed., 1994.

[40] Q. Wang *et al.*, *Technical Report on Building Robust Wireless LAN for Industrial Control with DSSS-CDMA Cellphone Network Paradigm*. `http://www-rtsl.cs.uiuc.edu/papers/dsss_cdma_tr_jnl.pdf`.

[41] A. Muqattash and M. Krunz, "Cdma-based mac protocol for wireless ad hoc networks," in *Proceedings of the 4th ACM Intl' Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003)*, Annapolis, Maryland, USA, 2003, pp. 153–164.

[42] R. Price and P. E. G. Jr., "A communication technique for multipath channels," *Proceedings of the IRE*, vol. 46, pp. 555–570, 1958.

[43] *DRCL J-Sim*. `http://www.j-sim.org`.

[44] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. (to be published by) Cambridge University Press, draft ed., 2004.

[45] *Cellular System Recommended Minimum Performance Standards for Full-Rate Speed Codes*. TIA/EIA/IS no.95, 1992.

[46] *IEEE Standard 802.11b*. 1999.

[47] *IEEE Standard 802.11a*. 1999.

[48] *IEEE Standard 802.11g*. 2003.

[49] A. R. S. Bahai and B. R. Saltzberg, *Multi-Carrier Digital Communications: Theory and Applications of OFDM*. Kluwer Academic/Plenum Publishers, 1999.

[50] L. Hanzo, W. Webb, and T. Keller, *Single- and multi-carrier quadrature amplitude modulation: principes and applications for personal communications, WLANs and broadcasting*. John Wiley & Sons, Ltd., 2000.

[51] J. Heiskala and J. Terry, *OFDM Wireless LANs: A Theoretical and Practical Guide*. Sams Publishing, 2002.

[52] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LAN's," in *Proceedings of the Conference on Communications Architectures, Protocols and Applications (ACM SIGCOMM 1994)*, London, UK, 1994, pp. 212–225.

[53] S. G. Wilson, *Digital Modulation and Coding*. Prentice Hall, 1996.

[54] *CDMA 2000 Series, Release A (2000)*. TIA/EIA/IS 2000 Series, Release A, 2000.

[55] *UMTS Forum*. `http://www.umts-forum.org`.

[56] *TD-SCDMA Forum*. `http://www.tdscdma-forum.org`.

[57] *QUALCOMM CDMA Technologies*. `http://www.cdmatech.com`.

[58] L. Korowajczuk, B. de Souza Abreu Xavier, A. M. F. Filho, *et al.*, *Designing cdma2000 Systems*. Wiley, 2004.

[59] *IEEE Standard 802.15.1*. 2005.

[60] J. C. Haartsen and S. Zurbes, "Bluetooth voice and data performance in 802.11 ds wlan environment," tech. rep., Ericsson SIG publication, 1999.

[61] Q. Wang, X. Liu, W. Chen, W. He, and M. Caccamo, "Building robust wireless lan for industrial control with dsss-cdma cellphone network paradigm," in *Proc. of the 26th IEEE International Real-Time Systems Symposium (RTSS 2005)*, Dec. 2005, pp. 3–14.

[62] A. K. Parekh, "A generalized processor sharing approach to flow control in integrated services network," Ph.D. dissertation, EECS Dept., M.I.T., Feb. 1992.

[63] J. C. R. Bennett *et al.*, "WF^2Q: Worst-case fair weighted fair queueing," *Proc. of INFOCOM'96*, pp. 120–128, 1996.

[64] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round robin," in *Proc. of SIGCOMM*, 1995, pp. 231–242.

[65] J. W. S. Liu, *Real-Time Systems*. Prentice Hall, 2000.

[66] M. Karol, M. Hluchyj, and S. Morgan, "Input versus output queueing on a space-division swtich," *IEEE Transactions on Communications*, vol. 35, pp. 1347–1356, Dec. 1987.

[67] N. W. McKeown, "Scheduling algorithms for input-queued cell switches," Ph.D. dissertation, EECS Dept., University of California at Berkeley, 1995.

[68] I. Elhanany, M. Kahane, and D. Sadot, "Packet scheduling in next-generation multiterabit networks," *IEEE Computer*, vol. 34, pp. 104–106, Apr. 2001.

[69] S. Gopalakrishnan, M. Caccamo, and L. Sha, "Switch scheduling and network design for real-time systems," in *Proc. of IEEE Real-Time and Embedded Technology and Applications (RTAS)*, Apr. 2006.

[70] Z. Deng and J. W.-S. Liu, "Scheduling real-time applications in an open environment," *Proc. of IEEE RTSS'97*, 1997.

[71] I. Shin and I. Lee, "Periodic resource model for compositional real-time guarantees," in *Proc. of the 24th IEEE International Real-Time Systems Symposium (RTSS 2003)*, Dec. 2003.

[72] G. Lipari and E. Bini, "Resource partitioning among real-time applications," *Proc. of ECRTS*, 2003.

[73] L. L. Peterson and B. S. Davie, *Computer Networks: A System Approach*. Margan Kaufmann, second ed., 2000.

[74] T.-W. Kuo and C.-H. Li, "A fixed-priority-driven open environment for real-time applications," *Proc. of IEEE RTSS'99*, 1999.

[75] R. Davis and A. Burns, "Hierarchical fixed priority preemptive scheduling," *Proc. of IEEE RTSS'05*, 2005.

[76] R. Davis and A. Burns, "Resource sharing in hierarchical fixed priority pre-emptive systems," *Proc. of IEEE RTSS'06*, 2006.

[77] T. Gonzalez and S. Sahni, "Open shop scheduling to minimize finish time," *Journal of the Association for Computing Machinery*, vol. 23, pp. 665–679, Oct. 1976.

[78] B. Fisher *et al.*, "Seeing, hearing, and touching: Putting it all together," *SIGGRAPH'04 Course*, 2004.

[79] M. Glencross *et al.*, "Exploiting perception in high-fidelity virtual environments," *SIGGRAPH'06 Course*, 2006.

[80] R. S. Raji, "Smart networks for control," *IEEE Spectrum*, vol. 31, pp. 49–55, June 1994.

[81] L. Sha, R. Rajkumar, and J. P. Lehoczky, "Real-time scheduling support in Futurebus+," in *Proceedings of the IEEE Real-Time Systems Symposium*, Dec. 1990, pp. 331–340.

[82] S. Gopalakrishnan, L. Sha, and M. Caccamo, "Hard real-time communication in bus-based networks," in *Proceedings of the IEEE Real-Time Systems Symposium*, Dec. 2004.

[83] J. Rexford, J. Hall, and K. G. Shin, "A router architecture for real-time communication in multicomputer networks," *IEEE Transactions on Computers*, vol. 47, pp. 1088–1101, Oct. 1998.

[84] C. Venkatramani and T. Chiueh, "Design and implementation of a real-time switch for segmented Ethernets," in *Proceedings of the International Conference on Network Protocols*, October 1997.

[85] D. Shah, P. Giaccone, E. Leonardi, and B. Prabhakar, "Delay bounds for combined input and output switches with low speedups," *Performance Evaluation*, vol. 55, no. 1-2, 2004.

[86] D. Shah, P. Giaccone, and E. Leonardi, "Throughput region of finite-buffered networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, Feb. 2007.

[87] S. Deb, D. Shah, and S. Shakkottai, "Fast matching algorithms for repetitive optimization: an application to switch scheduling," in *Proceedings of the Conference on Information, Sciences and Systems*, 2006.

[88] S.-T. Chuang, A. Goel, N. McKeown, and B. Prabhakar, "Matching ouput queueing with a combined input/output-queued switch," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1030–1039, June 1999.

[89] H.-I. Lee and S.-W. Seo, "Matching output queueing with a multiple input/output-queued switch," *IEEE/ACM Trans. on Networking*, vol. 14, pp. 121–132, February 2006.

[90] Q. Wang, S. Gopalakrishnan, X. Liu, and L. Sha, "A switch design for real-time industrial networks," in *(full paper accepted for publication) in Proceedings of the 14th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2008)*, Apr. 2008.

[91] X. Sheng *et al.*, "Energy based acoustic source localization," in *Information Processing in Sensor Networks: 2nd Intl. Workshop (IPSN 2003)*, vol. 2634 of *LNCS*, Springer, 2003, pp. 285–300.

[92] Q. Wang *et al.*, "Acoustic target tracking using tiny wireless sensor devices," in *Information Processing in Sensor Networks: 2nd Intl. Workshop (IPSN 2003)*, vol. 2634 of *LNCS*, Springer, 2003, pp. 642–657.

[93] Q. Wang, *Applying Lightning Protocol to Gunshot Localization.* `http://www-rtsl.cs.uiuc.edu/papers/technicalreports.html`, Nov. 2006.

[94] *MICA2 Wireless Measurement System Datasheet.* `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/6020-0042-04_A_MICA2.pdf`, Aug. 2003.

[95] *TinyOS a component-based OS for the networked sensor regime.* `http://webs.cs.berkeley.edu/tos/`, 2004.

[96] *RFM TR1000 916.50MHz Hybrid Tansceiver.* `http://today.cs.berkeley.edu/tos/hardware/design/data_sheets/RFM.pdf`, Apr. 2004.

[97] F. Calì *et al.*, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Transactions on Networking (TON)*, vol. 8, pp. 785–799, Dec. 2000.

[98] *Lightning Protocol Demo.* `http://www-rtsl.cs.uiuc.edu/papers/LightningDemo.html`, Apr. 2004.

[99] T.S.Rappaport, *Wireless communications: principles and practice (2nd Ed.).* Prentice Hall, 2004.

[100] X. Sheng, Y.-H. Hu, and P. Ramanathan, "Distributed particle filter with gmm approximation for multiple targets localization and tracking in wireless sensor network," in *4th International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, Apr. 2005, pp. 181–188.

[101] J. Aslam *et al.*, "Tracking a moving object with a binary sensor network," in *Proc. of the 1st Intl. Conf. on Embedded Networked Sensor Systems (SenSys'03)*, 2003, pp. 150–161.

[102] W. Kim, K. Mechitov, J.-Y. Choi, and S. Ham, "On target tracking with binary proximity sensors," in *4th International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, Apr. 2005, pp. 301–308.

[103] G. Simon *et al.*, "WSN-based shooter localization," in *Information Processing in Sensor Networks: 3rd Intl. Symp. (IPSN 2004)*, 2004. demo session.

[104] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, T. F. Abdelzaher, J. Hui, and B. Krogh, "Vigilnet: An integrated sensor network system for energy-efficient surveillance," *ACM Trans. Sen. Netw.*, vol. 2, no. 1, pp. 1–38, 2006.

[105] A. Arora, R. Ramnath, E. Ertin, P. Sinha, S. Bapat, V. Naik, V. Kulathumani, H. Zhang, H. Cao, M. Sridharan, S. Kumar, N. Seddon, C. Anderson, T. Herman, N. Trivedi, C. Zhang, M. Nesterenko, R. Shah, S. Kulkarni, M. Aramugam, L. Wang, M. Gouda, Y. ri Choi, D. Culler, P. Dutta, C. Sharp, G. Tolle, M. Grimmer, B. Ferriera, and K. Parker, "Exscal: Elements of an extreme scale wireless sensor network," in *Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05)*, 2005, pp. 102–108.

[106] S. Bapat, V. Kulathumani, and A. Arora, "Analyzing the yield of exscal, a large-scale wireless sensor network experiment," in *Proc. of the 13th IEEE International Conference on Network Protocols (ICNP'05)*, 2005, pp. 53–62.

[107] X. Liu, Q. Wang, L. Sha, and W. He, "Optimal QoS sampling frequency assignment for real-time wireless sensor networks," in *Proceedings of the 24th IEEE Real-Time Systems Symposium (RTSS 2003)*, Cancun, Mexico, Dec. 2003.

[108] B. Blum *et al.*, "An entity maintenance and connection service for sensor networks," in *Proc. of the 1st Intl. Conf. on Mobile Systems, Applications, and Services (MobiSys)*, 2003.

[109] W.-P. Chen, C.-J. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing (Special Issue on Sensor Networks)*, May 2004.

[110] S. Oh and S. Sastry, "Tracking on a graph," in *4th International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, Apr. 2005, pp. 195–202.

[111] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, 2004.

[112] I. Rhee, A. Warrier, M. Aia, and J. Min, "Z-mac: a hybrid mac for wireless sensor networks," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*, 2005, pp. 90–101.

[113] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proceedings of the 5th symposium on Operating systems design and implementation (OSDI'02)*, 2002, pp. 147–163.

[114] Q. Wang, R. Zheng, A. Tirumala, X. Liu, and L. Sha, "Lightning: A hard real-time, fast, and lightweight low-end wireless sensor election protocol for acoustic event localization," *IEEE Transactions on Mobile Computing*, vol. 7, pp. 570–584, May 2008.

[115] Q. Wang, R. Zheng, A. Tirumala, X. Liu, and L. Sha, "Lightning: A fast and lightweight acoustic localization protocol using low-end wireless micro-sensors," in *Proc. of the 25th IEEE International Real-Time Systems Symposium (RTSS 2004)*, Dec. 2004, pp. 371–381.

[116] Q. Wang, W. Shin, X. Liu, Z. Zeng, C. Oh, B. AlShebli, M. Caccamo, C. Gunter, E. Gunter, J. Hou, K. Karahalios, and L. Sha, "I-living: An open system architecture for assisted living," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics 2006 (SMC'06), Invited Paper*, vol. 5, 8-11 Oct. 2006, pp. 4268–4275.

[117] J. C. Hou, Q. Wang, B. K. AlShebli, L. Ball, S. Birge, M. Caccamo, C.-F. Cheah, E. Gilbert, C. A. Gunter, E. Gunter, C.-G. Lee, K. Karahalios, M.-Y. Nam, N. Nitya, C. Rohit, L. Sha, W. Shin, S. Yu, Y. Yu, and Z. Zeng, "Pas: A wireless-enabled, sensor-integrated personal assistance system for independent and assisted living," in *HCMDSS-MDPNP '07: Proceedings of the 2007 Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability*, 2007, pp. 64–75.

Curriculum Vitae

Qixin Wang was born in Shanghai, China, in 1977. He got his Bachelor of Engineering and Minor Bachelor of Economics degrees in 1999, and Master of Engineering degree in 2001, all from Tsinghua University, Beijing. He is currently pursuing the PhD degree in the Department of Computer Science, University of Illinois at Urbana-Champaign. His research interests include real-time/embedded systems and and their applications in industrial control, medicine, and assisted living. He is a student member of IEEE and ACM.

Publications

Journal Publications:

1. Qixin Wang, Rong Zheng, Ajay Tirumala, Xue Liu, and Lui Sha, "Lightning: A Hard Real-Time, Fast, and Lightweight Low-End Wireless Sensor Election Protocol for Acoustic Event Localization,"

in IEEE Transactions on Mobile Computing, volume 7, issue 5, pp. 570-584, May, 2008. (Demo Video Available at YouTube)(Selected as the Featured Article of IEEE TMC 2008 May Issue)

2. Qixin Wang, Xue Liu, Weiqun Chen, Lui Sha, and Marco Caccamo, "Building Robust Wireless LAN for Industrial Control with the DSSS-CDMA Cell Phone Network Paradigm," in IEEE Transactions on Mobile Computing, volume 6, issue 6, pp. 706-719, June, 2007. (A Note on Abuse of Terms: During the writing of the paper, we used the term "IEEE 802.15.4a/b/c" to refer to different PHY schemes of IEEE 802.15.4 for ease of narration. We were unaware at that time that IEEE 802.15.4 task groups are now using the term "IEEE 802.15.4a/b/c" for other purposes. Fortunately, the correct meanings of "IEEE 802.15.4a/b/c" for our paper are explicitly defined in Table 3. – Qixin Wang 8/21/2007)

3. Xue Liu, Qixin Wang, Wenbo He, Marco Caccamo, and Lui Sha, "Optimal Real-Time Sampling Rate Assignment for Wireless Sensor Networks," in ACM Transactions on Sensor Networks, volume 2, issue 2, pp. 263-295, May, 2006.

Conference, Workshop, and Other Publications during Ph.D. Program:

1. Qixin Wang, Sathish Gopalakrishnan, Xue Liu, and Lui Sha, "A Switch Design for Real-Time Industrial Networks," in Proceedings of the 14th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2008), pp. 367-376, April, 2008.

2. Qixin Wang, Xue Liu, Jennifer Hou, and Lui Sha, "GD-Aggregate: A WAN Virtual Topology Building Tool for Hard Real-Time and Embedded Applications," in Proceedings of the 28th IEEE Real-Time Systems Symposium (RTSS 2007), pp. 379-388, Tucson, Arizona, Dec. 3-6, 2007.

3. Zheng Zeng, Sammy Yu, Bedoor Alshebli, and Qixin Wang, "Personal Assist System: Use of Wireless Technologies Helping Elder People with Assisted Living," in ACM MobiCom 2007 Demo Session, Sep., 2007.

4. Jennifer Hou, Qixin Wang, Linda Ball, Stanley Birge, Marco Caccamo, Chin-Fei Cheah, Eric Gilbert, Carl Gunter, Elsa Gunter, Chang-Gun Lee, Karrie Karahalios, Min-Young Nam, Narasimhan Nitya, Chaudhri Rohit, Lui Sha, Wook Shin, Yang Yu, and Zheng Zeng, "PAS: A Wireless-Enabled, Sensor-Integrated Personal Assistance System for Independent and Assisted Living," in Proc. of Joint Workshop on High Confidence Medical Devices, Software, and Systems (HCMDSS) and Medical Device Plug-and-Play (MD PnP) Interoperability (HCMDSS/MD PnP'07), Boston, MA, June 25-27, 2007.

5. Mu Sun, Qixin Wang, and Lui Sha, "Building Reliable MD PnP Systems," in Proc. of Joint Workshop on High Confidence Medical Devices, Software, and Systems (HCMDSS) and Medical Device Plug-and-Play (MD PnP) Interoperability (HCMDSS/MD PnP'07), Boston, MA, June 25-27, 2007.

6. Qixin Wang, Wook Shin, Xue Liu, Zheng Zeng, Cham Oh, Bedoor K. Alshebli, Marco Caccamo, Carl A. Gunter, Elsa Gunter, Jennifer Hou, Karrie Karahalios, and Lui Sha, "I-Living: An Open System Architecture for Assisted Living," (invited paper) in Proc. of IEEE International Conference on Systems, Man, and Cybernetics (ICSMC'06), vol. 5, pp. 4268-4275, Oct. 8-11, 2006.

7. Qixin Wang, Xue Liu, Weiqun Chen, Wenbo He, and Marco Caccamo, "Building Robust Wireless LAN for Industrial Control with DSSS-CDMA Cellphone Network Paradigm," in Proc. of the 26th IEEE International Real-Time Systems Symposium (RTSS 2005), pp. 3-14, Miami, USA, Dec. 5-8, 2005.

8. Xue Liu, Rong Zheng, Jin Heo, Qixin Wang, and Lui Sha, "Timing Control for Web Server Systems Using Internal State Information," in Proc. of Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services (ICAS-ICNS 2005), Oct. 23-28, 2005.

9. Qixin Wang, Rong Zheng, Ajay Tirumala, Xue Liu, and Lui Sha, "Lightning: A Fast and Lightweight Acoustic Localization Protocol Using Low-End Wireless Micro-Sensors," in Proc. of the 25th IEEE International Real-Time Systems Symposium (RTSS 2004), pp. 371-381, Lisbon, Portugal, Dec. 5-8, 2004.

10. Xue Liu, Qixin Wang, Lui Sha and Wenbo He, "Optimal QoS Sampling Frequency Assignment for Real-Time Wireless Sensor Networks," in Proc. of the 24th IEEE International Real-Time Systems Symposium (RTSS 2003), pp. 308-319, Cancun, Mexico, Dec. 3-5, 2003.

11. Qixin Wang, Wei-Peng Chen, Rong Zheng, Kihwal Lee, and Lui Sha, "Acoustic Target Tracking Using Tiny Wireless Sensor Devices," in Proc. of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN'03), Poster Session, Lecture Notes in Computer Science 2634, Palo Alto, CA, USA: Springer, pp. 642-657, Apr. 22-23, 2003.

Selected Publications before Entering Ph.D. Program:

1. Wang, Qi-Xin; Li, Yi; Dong, Li; Nie, Yu; Wang, Ke-Hong. "Incremental and distributed web page clustering algorithms PG+ and PG++," in Ruan Jian Xue Bao/Journal of Software, v 13, n 8, August, 2002, pp. 1500-1507 Language: Chinese (Ei Compendex indexed).

2. Qixin Wang, Study and Implementation of Clustering Algorithms for Web Site Personalization, Master's Thesis, 2001 Language: Chinese.

3. Qixin Wang, Xinghua An, Xiaoge Wang, Kehong Wang. "JiniSolve: The Primary Architecture and Experiences of a Jini Based Metacomputing Framework," in Ji Suan Ji Gong Cheng Yu Ying Yong/Computer Engineering and Applications, n 21, 2001 Language: Chinese.

4. Qixin Wang, Xinghua An, Xiaoge Wang, Kehong Wang. "Overall Introduction to Jinisolve: a Jini Based Grid Computing Framework," in Proc. of the 2nd International Conference on Parallel and Distributed Computing, Applications, and Technologies (PDCAT 2001), July 9-11, 2001.

Reviewer For

1. IEEE Real-Time Systems Symposium (RTSS) 2007, 2006, 2004, 2003
2. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS) 2008, 2006, 2004
3. EuroMicro Conference on Real-Time Systems (ECRTS) 2007, 2005
4. International Conference on Information Processing in Sensor Networks (IPSN) 2007, 2004
5. Joint Workshop on High Confidence Medical Devices, Software, and Systems (HCMDSS) and Medical Device Plug-and-Play (MD PnP) Interoperability 2007
6. IEEE Wireless Communications and Networking Conference (WCNC) 2006
7. IEEE Transactions on Computers
8. IEEE Transactions on Wireless Communications
9. IEEE Transactions on Parallel and Distributed Systems
10. ACM Transactions on Sensor Networks
11. Computer Networks Journal (COMNET), Elsevier
12. Real-Time Systems Journal, Sprint

Awards

1. Vodafone Fellowship (by Vodafone and UIUC), in 2005, 2003
2. UIUC Graduate College Conference Travel Grant (by UIUC), in 2005

3. IPSN03 Student Travel Grant (by IPSN), in 2003

4. Friend of Tsinghua Motorola Scholarship (by Motorola and Tsinghua Univ.), in 2000

5. Honored Graduate of Tsinghua University (by Tsinghua University), in 1999

6. Computer World Scholarship (by Chinese Computer World Magazine), in 1998

7. Outstanding Student Scholarship (by Tsinghua University), in 1997, 1996, 1995, 1994

8. Outstanding Freshman Scholarship (by Tsinghua University), in 1994